POLI TÉCNICO GUARDA

Escola Superior de Tecnologia e Gestão

MÁQUINA PREDITIVA K-MEANS COM NÚMERO MÁGICO

GESTÃO DE PROJETOS - TESP ANÁLISE DE DADOS

Ricardo Dias Ferreira
Janeiro / de 2023

POLI TÉCNICO GUARDA

Escola Superior de Tecnologia e Gestão

MÁQUINA PREDITIVA K-MEANS COM NÚMERO MÁGICO

GESTÃO DE PROJETOS - TESP ANÁLISE DE DADOS

Professor(a): Celestino Pereira Gonçalves

Coordenador(a): Paulo Vieira

Ricardo Dias Ferreira - 1706995

Janeiro 2023

RESUMO

O presente relatório foi elaborado como parte integrante da unidade curricular Gestão de Projeto, leccionadas no âmbito do Curso Técnico Profissional de Análise de Dados pretendendo demostrar as atividades relacionadas ao projeto desenvolvido no decorrer das aulas. Constam neste relatório oito capítulos, cada um deles dedicado a pontos importantes relacionados ao assunto do projeto desenvolvido e da respectiva unidade curricular, e ainda dois anexos que auxiliam de forma diferenciada no entendimento de alguns pontos. No primeiro capítulo temos uma introdução ao que será desenvolvido. No segundo capítulo faço uma extenuada pesquisa sobre o estado da arte para os assuntos abordados, na qual encontrei bastante material sobre arquivos preditivos feitos com o k-means e também outros artigos relativos ao chamado número mágico. No terceiro capítulo faço referência as tecnologias e algumas definições de algoritmos que utilizo na elaboração do projeto. No quarto capítulo faço menção da abordagem ágil, que foi a metodologia utilizada, mostro as etapas do desenvolvimento e apresento o termo de abertura que construímos no decorrer das aulas. No quinto capítulo temos a análise de requisito, também essa parte do projeto desenvolvida nas aulas de gestão de projeto, com diagrama de contexto, casos de uso etc.; e também falo do impacto do novo Regulamento Geral sobre a Proteção de Dados (RGPD) nesse trabalho. No sexto capítulo temos a parte de implementação do projeto, onde trago uma caracterização do principal algoritmo utilizado (k-means), uma explicação sobre a análise de cluster e ainda a caracterização do três dataset's utilizados no projeto. No sétimo capítulo temos a verificação e validação do trabalho com testes, isso é demonstrado com imagens de códigos e resultados dos mesmos após execução. Por fim temos a conclusão do trabalho realizado e ainda fazemos referências ao que pode ser realizado a partir do mesmo.

Palavras-chave: K-means. Projeto. Dataset. Agrupamento. Algoritmo.

ABSTRACT

This report was prepared as an integral part of the Project Management curricular unit, taught within the Professional Technical Course of Data Analysis, intending to demonstrate the activities related to the project developed during the classes. This report contains eight chapters, each one dedicated to important points related to the subject of the project developed and the respective curricular unit, and two annexes that help in a different way in the understanding of some points. In the first chapter we have an introduction to what will be developed. In the second chapter I make an exhaustive research on the state of the art for the addressed subjects, in which I found a lot of material on predictive files made with k-means and also other articles related to the so-called magic number. In the third chapter I make reference to the technologies and some definitions of algorithms that I use in the elaboration of the project. In the fourth chapter I mention the agile approach, which was the methodology used, I show the stages of development and present the opening term that we built during the classes. In the fifth chapter we have the requirement analysis, also this part of the project developed in the project management classes, with context diagram, use cases etc.; and I also talk about the impact of the new General Data Protection Regulation (GDPR) on this work. In the sixth chapter we have the project implementation part, where I bring a characterization of the main algorithm used (k-means), an explanation about the cluster analysis and also the characterization of the three datasets used in the project. In the seventh chapter we have the verification and validation of the work with tests, this is demonstrated with images of codes and their results after execution. Finally, we have the conclusion of the work done and we also make references to what can be done from it.

Keywords: K-means. Project. Dataset. Grouping. Algorithm.

ÍNDICE

RESUMO ..... **iii**

ABSTRACT ..... **iv**

ÍNDICE DE FIGURAS ..... **vi**

LISTA DE ABREVIATURS E SIGLAS ..... **viii**

1. Introdução **9**

2. Estado da arte **10**

3. Tecnologias utilizadas **12**

4. Metodologia **14**

Método utilizado 14

Etapas do desenvolvimento (Gráfico de Gantt) 15

Termo de abertura 15

5. Análise de requisitos **17**

Impacto da RGPD no projeto 22

6. Caracterização da máquina preditiva **23**

Caracterização do algoritmo utilizado 23

O que é Análise de Cluster? 23

Método K-means 24

7. Caracterização dos dataswt's utilizados **26**

Avaliações de hotéis (Hotels Reviews(booking.com)) 26

Clientes_Shopping (Mall_Customers) 26

Consumo_de_energia_doméstica (household_power_consumption) 27

8. Análise e exploratória dos dados **29**

Avaliações de hotéis (Hotels Reviews(booking.com)) 29

Clientes_Shopping (Mall_Customers) 37

Consumo_de_energia_doméstica (household_power_consumption) 46

9. Discussão de resultados **56**

Como descobrir o número ideal de clusters? 56

10. Conclusão e trabalhos futuros **61**

Referencias **62**

ÍNDICE DE FIGURAS

Figura 1 - Cronog 15
Figura 2 - Suporte 15
Figura 3 - DiagContexto 17
Figura 4 - DiagCaso 18
Figura 5 - Imports (booking.com) 29
Figura 6 - carreg (booking.com) 29
Figura 7 - Head (booking.com) 29
Figura 8 - dim (booking.com) 30
Figura 9 - Tipo (booking.com) 30
Figura 10 - Info (booking.com) 30
Figura 11 - Iloc (booking.com) 31
Figura 12 - Head (booking.com) 31
Figura 13 - Valores (booking.com) 31
Figura 14 - Isnull (booking.com) 32
Figura 15 – pca (booking.com) 32
Figura 16 - Func7+-2 (booking.com) 33
Figura 17 – Execut func (booking.com) 34
Figura 18 - Names (booking.com) 34
Figura 19 - Nova variável (booking.com) 34
Figura 20 - Novo dataset (booking.com) 34
Figura 21 – Med/pont (booking.com) 35
Figura 22 - Med/conf (booking.com) 35
Figura 23 - Méd/cb (booking.com) 36
Figura 24 - count (booking.com) 36
Figura 25 – Dispersão (booking.com) 36
Figura 26 - Imports (Mall_Customers) 37
Figura 27 - Carreg (Mall_Customers) 37
Figura 28 - Head (Mall_Customers) 37
Figura 29 - dim (Mall_Customers) 38
Figura 30 - Tipos (Mall_Customers) 38
Figura 31 - Info (Mall_Customers) 38
Figura 32 - Desc (Mall_Customers) 39
Figura 33 - Iloc (Mall_Customers) 39
Figura 34 - Visual (Mall_Customers) 39
Figura 35 - Values (Mall_Customers) 40
Figura 36 - Missing (Mall_Customers) 40
Figura 37 – Func7+-2 (Mall_Customers) 40
Figura 38 - Exec função (Mall_Customers) 41
Figura 39 - Names (Mall_Customers) 41
Figura 40 - Cluster_map (Mall_Customers) 42
Figura 41 - Novo Dataset (Mall_Customers) 42
Figura 42 - Sep cluster (Mall_Customers) 42
Figura 43 - Med/idade (Mall_Customers) 43
Figura 44 - Med/renda (Mall_Customers) 43
Figura 45 - Med/pont (Mall_Customers) 43
Figura 46 - Count (Mall_Customers) 44
Figura 47 - Pct/cluster (Mall_Customers) 44
Figura 48 - CodDisp (Mall_Customers) 44
Figura 49 - GráfDisp (Mall_Customers) 45
Figura 50 - Imports (HPC) 46

FIGURA 51 - READ (HPC) 46
FIGURA 52 - HEAD (HPC) 46
FIGURA 53 - DIM (HPC) 47
FIGURA 54 - TIPOS (HPC) 47
FIGURA 55 - INFO (HPC) 47
FIGURA 56 - MISSING (HPC) 48
FIGURA 57 - SUM (HPC) 48
FIGURA 58 - ILOC (HPC) 48
FIGURA 59 - ANY (HPC) 48
FIGURA 60 - SUM2 (HPC) 49
FIGURA 61 - DF (HPC) 49
FIGURA 62 - VALUES (HPC) 50
FIGURA 63 - AMOSTRA (HPC) 50
FIGURA 64 - SHAPE (HPC) 50
FIGURA 65 - PCA (HPC) 51
FIGURA 66 - FUNCAO 7+-2 (HPC) 51
FIGURA 67 - EXECFUNC (HPC) 52
FIGURA 68 - NAMES (HPC) 52
FIGURA 69 - CLUSTERMAP (HPC) 53
FIGURA 70 - NOVODATASET (HPC) 53
FIGURA 71 - SEPCLUSTER (HPC) 53
FIGURA 72 - MEDIA (HPC) 54
FIGURA 73 - COUNT (HPC) 54
FIGURA 74 - PCT (HPC) 54
FIGURA 75 - CODDISP (HPC) 55
FIGURA 76 – GRAFDISP (HPC) 55
FIGURA 77 - CODELBOW (HPC) 56
FIGURA 78 - GRAFELBOW (HPC) 57
FIGURA 79 - PCA (ANEXO) 57
FIGURA 80 - SEP (ANEXO) 58
FIGURA 81 - MEDIA (ANEXO) 58
FIGURA 82 - COUNT (ANEXO) 59
FIGURA 83 - PCT (ANEXO) 59
FIGURA 84 - CODDISP (ANEXO) 60
FIGURA 85 - GRAFDISP (ANEXO) 60

LISTA DE ABREVIATURS E SIGLAS

PCA	Principal Component Analysis
RGPD	Regulamento Geral sobre a Proteção de Dados
IPG	Instituto Politécnico da Guarda
ESTG	Escola Superior de Tecnologia e Gestão
EU	European Union

1. Introdução

Nesse projeto vou desenvolver uma Máquina Preditiva utilizando banco de dados que foram gerados por humanos, esses dados serão tratados em programas específicos. O principal algoritmo utilizado no desenvolvimento será o k-means, trata-se de um tipo de aprendizagem não supervisionada que forma automaticamente agrupamentos de coisas semelhantes. Pode-se agrupar quase tudo, e quanto mais semelhantes forem os itens no cluster, melhores serão os agrupamentos.

"A memória operacional pode chegar a albergar sete (mais ou menos dois) sequências de informações". **George Armitage Miller**. George Miller ao estudar a quantidade de informação que o sistema cognitivo é capaz de processar, chegou ao número que sempre lhe recorria, sete, as vezes um pouco mais, as vezes um pouco menos. O número "mágico" apareceu nalgumas experiências para determinar o alcance do critério absoluto e com quanta precisão somos capazes de distinguir vários estímulos diferentes. Miller percebeu em outros estudos que com o aumento das variáveis, a precisão também descia ligeiramente. O artigo em que Miller trouxe essas informações transformou-se numa das obras fundamentais da psicologia cognitiva e do estudo da memória operacional (a capacidade de recordar e utilizar informações durante um período limitado). O projeto traz o desafio de fazer pesquisas de como desenvolver a máquina com o algoritmo k-means, encontrar a forma de verificar qual o melhor número de conjuntos, assim utilizando o número mágico.

Para o desenvolvimento do projeto vou contar com a ajuda dos principais motivadores para existência do mesmo, o professor **Paulo Vieira** para assuntos relacionados ao algoritmo, e com a ajuda do professor **Celestino Gonçalves** para assuntos relacionados a Gestão de Projetos.

O objetivo desse projeto é desenvolver uma Máquina Preditiva usando o algoritmo k-means em conjunto com o número mágico (7+-2) e a partir dos resultados construir outros *dataset's* para aprendizagem humana.

Esse relatório tem a intenção de apresentar um resumo das atividades realizadas bem como informar os dados e resultados coletados com elas. Sua estrutura apresenta: título, onde indico o assunto do projeto; introdução, onde temos um esclarecimento sucinto dos fundamentos teóricos nos quais está baseado o projeto; desenvolvimento, temos alguns capítulos e em cada um deles abordamos o tema, ora voltados a parte relacionada a gestão de projetos, ora voltados para a parte prática com o algoritmo k-means; conclusão, onde a medida de satisfação com o projeto e indicamos possíveis trabalhos futuros; referências, onde cito fontes que nos ajudaram e podem ser fontes de pesquisas futuras e/ou verificação; por fim, temos alguns anexos que abordam outras possibilidades e reafirmam escolhas.

2. Estado da arte

Com um crescimento explosivo em dados não estruturados e estruturados, as organizações buscam formas de inovação e travessias da análise e da ciência de dados; A interpretação de dados e a descoberta de nova informação através dos mesmos, conduz muitas vezes ao encontro de uma solução para problemas da atualidade, a aglomeração de dados, conhecida como clustering, é uma das tarefas mais comuns na área de análise de dados (statplace, 2022).

Máquinas preditivas são feitas com a ajuda de inteligência artificial, utilizando-se de vários algoritmos. Neste trabalho tenho o objetivo de principalmente utilizar um algoritmo de procura de padrões nos dados de modo a formar grupos os algoritmos de clusterização são utilizados com muita frequência em aplicações que necessitem disse. Algoritmos de Clusterização dividem os dados em grupos úteis ou significativos, nos quais a similaridade intra-cluster é maximizada e a similaridade inter-cluster é minimizada. Estes clusters descobertos podem ser usados para explicar as características da distribuição dos dados subjacentes e assim servir como base para várias técnicas de análise e mineração de dados. O K-means, algoritmo que vou usar, utiliza o conceito de centróides como protótipos representativos dos grupos, onde o centróide representa o centro de um grupo, sendo calculado pela média de todos os objetos do grupo. Esse algoritmo já foi aplicado em conjuntos de dados relacionado a comparação de hábitos para escolha de clientes, em dados biomédicos, perfil de consumo, etc.

George Miller tomou conta do nosso imaginário numa aula do semestre passado do Tesp de análise de dados, através do seu trabalho mais famoso: O número mágico sete, mais ou menos dois: alguns limites na nossa capacidade de processar informação. Nesse artigo, Miller dá ao número 7, a capacidade mágica de permitir que a nossa memória a curto prazo, tenha a capacidade de armazenar listas de letras, palavras, números ou qualquer tipo de itens discretos, na quantidade referente a esse número mágico (mais ou menos dois).

Sobre o estado da arte para esse assunto, encontrei bastante material sobre arquivos preditivos feitos com o k-means (*Vania_Gomes, 2013; Jose Mesdes, 2017; Dimitri Rodrigues, 2022*) e também outros artigos relativos ao chamado número mágico(*Michael Hotchkiss*, 2012; Maestrovirtuale/Psicologia/Biografias: biografia de um pioneiro da psicologia cognitiva, 2022; George Armitage Miller, O número mágico sete, mais ou menos dois: algumas limitações em nossa capacidade de processar informações de 1956), mas não encontrei nenhum trabalho com os dois juntos com o intuito de posteriormente esse trabalho ser usado para aprendizagem humana. Devido a isso elaboração desse projeto se faz ainda mais necessária, inovadora, e traz a possibilidade de um grande contributo.

Dado um conjunto de observações, o agrupamento K-means tentará agrupar as observações em um número pré-especificado de “k” clusters distintos e não

sobrepostos. É nesse momento que entraremos com o 7+-2; vamos testar a formação desses conjuntos com o “k” assumindo essas cinco possibilidades e escolheremos a que tem melhor taxa de acerto.

3. Tecnologias utilizadas

PyCharm

É um ambiente de desenvolvimento integrado usado para programação em Python.

Visual Studio Code

É um editor de código-fonte desenvolvido pela Microsoft

Github

É uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git. Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou código aberto de qualquer lugar do mundo.

Colab

O Colaboratory ou "Colab" é um produto do Google Research, área de pesquisas científicas do Google. O Colab permite que qualquer pessoa escreva e execute código Python pelo navegador e é especialmente adequado para aprendizado de máquina, análise de dados e educação.

Scikit-learn

É uma biblioteca de aprendizado de máquina de código aberto para a linguagem de programação Python.

PCA (Principal Component Analysis**)**

Uma das técnicas mais utilizadas na redução de dimensionalidade é um método estatístico designado por Principal Component Analysis (PCA). O PCA é caracterizado por identificar as dimensões ao longo das quais os dados se encontram mais dispersos. Desta forma, conseguimos identificar as dimensões que melhor diferenciam o conjunto de dados em análise, ou seja, os seus componentes principais. É um procedimento matemático que utiliza uma transformação ortogonal (ortogonalização de vetores) para converter um conjunto de observações de variáveis possivelmente correlacionadas num conjunto de valores de variáveis linearmente não correlacionadas chamadas de **componentes principais**. O número de componentes principais é sempre menor ou igual ao número de variáveis originais. Os componentes principais são garantidamente independentes apenas se os dados forem normalmente distribuídos (conjuntamente). O PCA é sensível à escala relativa das variáveis originais. Dependendo da área de aplicação, o PCA é também conhecido como transformada de Karhunen-Loève (KLT) discreta, transformada de Hotelling ou decomposição ortogonal própria (POD).

O PCA foi inventado em 1901 por Karl Pearson. Agora, é mais comumente usado como uma ferramenta de Análise Exploratória de Dados e para fazer modelos preditivos. PCA pode ser feito por decomposição em autovalores (Valores Próprios) de uma matriz covariância, geralmente depois de centralizar (e normalizar ou usar pontuações-Z) a matriz de dados para cada atributo.

Usando esta técnica, é possível realçar as semelhanças e diferenças neles existentes através da identificação de padrões. Quando identificados os padrões no conjunto, o número de dimensões a analisar pode ser reduzido sem que haja uma perda significativa de informação, pois o foco recai sobre a análise das dimensões principais que caracterizam o conjunto de dados.

O PCA é matematicamente definido como uma transformação linear ortogonal que transforma os dados para um novo sistema de coordenadas de forma que a maior variância por qualquer projeção dos dados fica ao longo da primeira coordenada (o chamado *primeiro componente*), a segunda maior variância fica ao longo da segunda coordenada, e assim por diante.

Microsoft Office

Processar texto, tabelas e apresentação do Microsoft Office

Ferramentas de pesquisa

Google, YouTube, Sites, etc.

4. Metodologia

Método utilizado

Nesse projeto decidi aplicar uma abordagem ágil pois, a abordagem ágil tem demonstrado ser melhor aplicada para entregas de software, onde a falta de fisicalidade e a variação de requisitos é tratada através da sua abordagem leve para a execução do projeto.

Como um projeto ágil seguimos o seguinte modelo de desenvolvimento:

- Estabelecem-se os fundamentos do projeto:

  A visão do negócio para o projeto;

  Os objetivos;

  Uma lista dos requisitos/funcionalidades;

- O projeto foi dividido em parcelas;
- Foram desenvolvidas as funcionalidades;
- As funcionalidades foram testadas e integradas;

Como num desenvolvimento ágil, cada aspeto do desenvolvimento do projeto – requisitos, desenho, arquitetura, construção, testes, etc. – foi continuamente revisitado ao longo do ciclo de vida do desenvolvimento.

A cada 2 semanas reavaliamos o projeto, o que possibilitou a condução numa direção diferente, quando necessário.

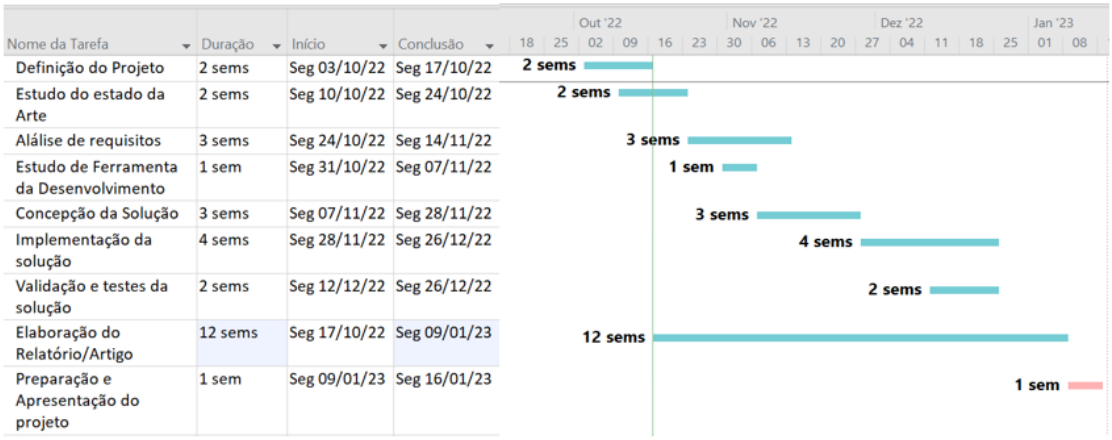
A combinação dos ciclos de desenvolvimento curtos e de duração fixa (iterações de 2 semanas) e da priorização dos requisitos possibilitou uma redução do risco global e distribuiu de modo uniforme o risco ao longo de todas as iterações.

Nessa abordagem ágil os processos de desenvolvimento são divididos em iterações de curta duração (sprints) e, como houve uma maior abertura para alterações aos requisitos, consumiu-se menos tempo no planeamento. Os princípios ágeis enfatizam um produto a funcionar, ao invés de uma documentação detalhada e abrangente. Por isso, na abordagem ágil começamos mais rapidamente a execução e gastamos menos tempo no planejamento.

Um aspeto fundamental da abordagem ágil é a adaptabilidade. A equipe de projeto não olhou apenas para o produto à medida que foi desenvolvido, ela também observou o processo. No final de cada iteração, a equipa realizou uma sessão de lições aprendidas, referida frequentemente como uma retrospectiva. O objetivo da sessão foi rever o produto e o processo. A equipa discutiu como o projeto está sendo desenvolvido e que alterações nos procedimentos melhorariam o seu desempenho.

Etapas do desenvolvimento (Gráfico de Gantt)

Durante todo o processo de desenvolvimento pretende-se fazer reuniões com os professores orientadores para acompanhamento e auxílio na elaboração do projeto, na imagem abaixo temos o cronograma que seguimos (Figura 1).



Nome da Tarefa	Duração	Início	Conclusão
Definição do Projeto	2 sems	Seg 03/10/22	Seg 17/10/22
Estudo do estado da Arte	2 sems	Seg 10/10/22	Seg 24/10/22
Alálise de requisitos	3 sems	Seg 24/10/22	Seg 14/11/22
Estudo de Ferramenta da Desenvolvimento	1 sem	Seg 31/10/22	Seg 07/11/22
Concepção da Solução	3 sems	Seg 07/11/22	Seg 28/11/22
Implementação da solução	4 sems	Seg 28/11/22	Seg 26/12/22
Validação e testes da solução	2 sems	Seg 12/12/22	Seg 26/12/22
Elaboração do Relatório/Artigo	12 sems	Seg 17/10/22	Seg 09/01/23
Preparação e Apresentação do projeto	1 sem	Seg 09/01/23	Seg 16/01/23

Figura 2 - Gsntt

Fonte: elaborado pelo autor.

Termo de abertura

Matriz de partes interessadas (Figura 2).

Partes interessadas no projeto

- Prof. Paulo Vieira
- Prof. Celestino Gonçalves
- Ricardo Ferreira
- Acadêmicos que terão acesso ao projeto
- Instituto Politécnico da Guarda - ESTG

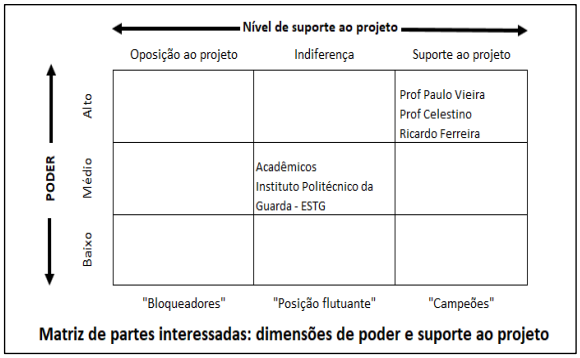


Figura 3 – Suporte

Fonte: elaborado pelo autor.

Objetivo do projeto

- Construir um dataset com resultados de pre
  means), a partir de outros três dataset que contenham dados com decisões humanas.
  - Data início 30/10/2022
  - Data término 09/01/2023

Sponsor do projeto

- Instituto Politécnico da Guarda

Dono do projeto

- Paulo Vieira

Gestor do projeto

- Ricardo Ferreira

Consultor de gestão do projeto

- Celestino Gonçalves (será consultado em caso de dúvidas sobre a gestão do projeto)

Gestão das operações

- Ricardo Ferreira

Restrições ao projeto

- Prazo - Gant
- Custo – Muito estudo e pesquisas
- Qualidade – Professores

Justificação empresarial do projeto

- analise o desempenho do comércio em determinado ano.

Órgãos reguladores

- **Autoridade Nacional de Proteção de Dados** (ANPD)
- Lei Geral de Proteção de Dados (LGPD)

Grupos de interesses especiais

- Segmentos empresariais
    - Podem após o modelo estar pronto e pedirem que se faça uma análise da área de atuação
- Órgãos da administração pública
    - Podem após o modelo estar pronto e pedirem que se faça uma análise da área de atuação

Requisito para execução do projeto

- Aprender a trabalhar com o algoritmo k-means
- Aprender a trabalhar com as ferramentas que serão utilizadas no projeto
- Conseguir conjunto de dados, com dados que são resultado de escolhas humanas para usar no projeto
- Acesso a internet para utilizar ferramentas e fazer pesquisas

5. Análise de requisitos

Diagrama de contexto (Figura 3).

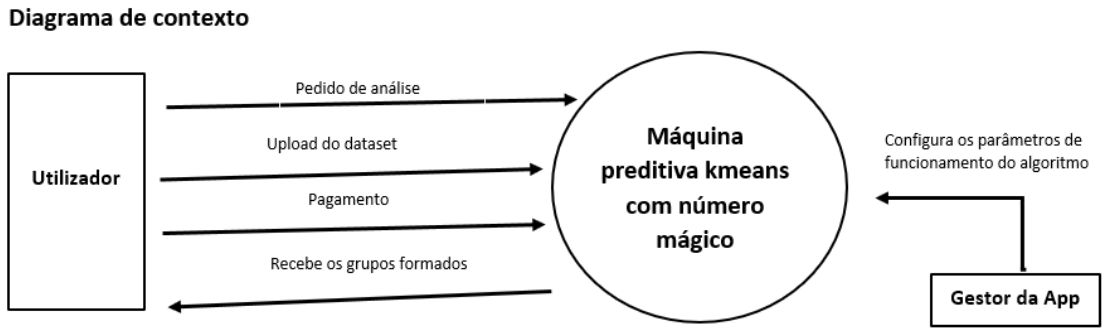


Figura 4 - DiagContexto

Fonte: elaborado pelo autor.

Atores e casos de uso (Tabela 1).

Tabela 2

ATOR	CASO DE USO	DESC CASO DE USO
Utilizador	Pedido de análise	O objetivo do caso de uso é que o utilizador faça um pedido de análise de cluster
Utilizador	Upload do dataset	O objetivo do caso de uso é que o utilizador faz o upload do dataset que deseja que seja analisado
Utilizador	Pagamento	O objetivo do caso de uso é que o utilizador faz o pagamento do valor da operação
Utilizador	Recebe os grupos formados	O objetivo do caso de uso é a máquina preditiva devolve o número o melhor agrupamento do dataset que recebeu
Gestor da App	Configuração dos parâmetros de funcionamento do algoritmo	O objetivo do caso de uso é que quando necessário o programador faça a configuração dos parâmetros de funcionamento da máquina preditiva

Diagrama de casos de uso (Figura 4).

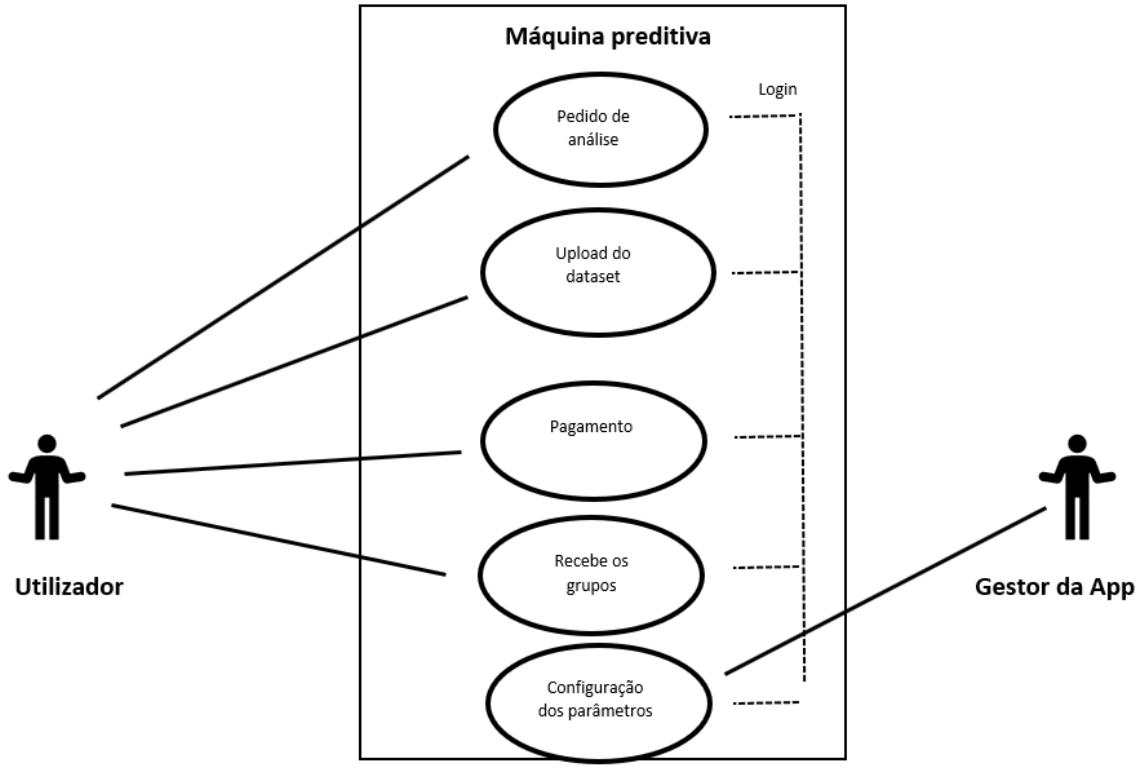


Figura 5 - DiagCaso

Fonte: elaborado pelo autor.

Descrição dos casos de uso "predição de grupos"

Nome: **Pedido de análise**

Descrição: o objetivo é que o utilizador faça um pedido de análise cluster para determinado dataset

Pré-condição: para executar o caso de uso, o ator tem de efetuar um login válido

Caminho principal:

Passo 1 – o ator seleciona a opção "pedido de análise" no menu de funções

Passo 2 – o sistema exibe a interface para pedir a caracterização do dataset

Passo 3 – o ator fornece a dimensão do dataset

Passo 4 - o ator fornece o tipo de dados do dataset

Passo 5 - o ator confirma o pedido de análise

Passo 6 – o sistema valida o pedido de análise

Caminho alternativo:

Passo 5a – o ator não confirma o pedido de análise

Passo 6a – o sistema não valida o pedido análise

Nome: **Upload do dataset**

Descrição: o objetivo é que o utilizador faça o upload do dataset que pretende que seja feita a análise

Pré-condição: para executar o caso de uso, o sistema precisa ter validado o pedido de análise.

Caminho principal:

Passo 1 – o ator clica no botão upload

Passo 2 – o sistema exibe a interface com a opção para anexo

Passo 3 – o ator clica na opção anexo

Passo 4 – o sistema abre uma pasta de navegação para que o ator escolha o arquivo que deseja anexar

Passo 5 – o ator seleciona o arquivo que corresponde ao dataset de que deseja a análise

Passo 6 – o sistema ativa a opção enviar arquivo

Passo 7 – o ator clica em enviar arquivo

Passo 8 – o sistema recebe o arquivo e verifica a dimensão e os tipos de dados

Passo 9 – o sistema valida o dataset e mostra a opção de enviar o arquivo para análise

Passo 10 – o ator clica em enviar arquivo para análise

Passo 11 – o arquivo é enviado a máquina preditiva

Caminho alternativo:

Passo 9a – o sistema não valida o dataset

Passo 9b - o sistema volta para o passo 2

Caminho alternativo:

Passo 10a – o ator não clica em enviar arquivo para análise (clica em cancelar)

Passo 10b – o sistema finaliza a operação

Nome: **Pagamento**

Descrição: o objetivo é que o utilizador faça o pagamento da fatura para que a análise seja feita

Pré-condição: para executar o caso de uso, o ator precisa ter enviado um arquivo para análise

Caminho principal:

Passo 1 – o sistema exibe a interface com a opção pagamento

Passo 2 – o ator clica na opção pagamento

Passo 3 – o sistema exibe a interface o valor da análise

Passo 4 – o ator confirma o valor da análise e clica em seguir

Passo 5 – o sistema exibe a interface os dados para pagamento e a opção seguir com análise

Passo 6 – o ator após realização do pagamento clica em seguir com análise

Passo 7 – o sistema confirma a realização pagamento

Passo 8 – o sistema realiza a análise de cluster

Caminho alternativo:

Passo 4a – o ator não confirma o valor da análise

Passo 4b – o ator clica em cancelar

Caminho alternativo:

Passo 7a – o sistema não confirma a realização do pagamento

Passo 7b – o sistema exibe na interface a informação “pagamento não confirmado”

Passo 7c – o sistema volta para o passo 5

Nome: **Recebe os grupos**

Descrição: O objetivo do caso de uso é o sistema devolve o número do melhor agrupamento do dataset que recebeu

Pré-condição: para executar o caso de uso, o sistema precisa chegado ao passo 8 do caso de uso “pagamento”.

Caminho principal:

Passo 1 – o sistema exibe na interface a opção visualizar análise

Passo 2 – o ator clica em visualizar análise

Passo 3 – o sistema exibe na interface a análise e dois botões: “baixar” e “pedir edição de parâmetros”.

Passo 4 – o ator clica em baixar

Passo 5 – o sistema abre uma pasta de navegação onde será feito o armazenamento do arquivo

Passo 6 – o sistema mostra na interface a opção encerrar

Passo 7 – o ator clica em encerrar

Passo 8 – a operação é encerrada

Caminho alternativo:

Passo 4a – o ator clica em pedir configuração de parâmetros

Passo 4b – o sistema exibe na interface a opção “número de cluster para análise”

Passo 4c – o ator digita o número de cluster que quer que a análise seja feita

Passo 4d – o sistema envia para o gestor do App o pedido de edição de parâmetros e o número de cluster

Pós-condição: o sistema salva o dataset e o resultado da operação.

Nome: **Configuração dos parâmetros**

Descrição: O objetivo do caso de uso é que quando necessário o programador faça configuração dos parâmetros de funcionamento da máquina preditiva

Pré-condição: para executar o caso de uso, o ator precisa receber do sistema o pedido de configuração de parâmetros e o número de cluster

Caminho principal:

Passo 1 – o ator recebe do sistema o pedido de edição de parâmetros e o número de cluster

Passo 2 – o sistema exibe na interface a opção editar parâmetros

Passo 3 – o ator clica em configurar parâmetros

Passo 4 – o sistema exibe na interface a opção número de cluster

Passo 5 - o ator digita o número de cluster

Passo 6 – o sistema exibe na interface a opção reiniciar análise

Passo 7 – o ator clica na opção reiniciar análise

Passo 8 – o sistema retorna ao início do caso de uso “Recebe os grupos”

Impacto da RGPD no projeto

Neste trabalho foi aplicado o RGPD desde a conceção e por defeito.

O novo Regulamento Geral sobre a Proteção de Dados (“RGPD”), aprovado pelo Parlamento Europeu e pelo Conselho Europeu, considera um direito fundamental, independentemente da nacionalidade ou local de residência, a proteção das pessoas singulares relativamente ao tratamento dos seus dados pessoais

Através deste regulamento pretende-se:

- Assegurar a defesa dos direitos e liberdades fundamentais das pessoas singulares;
- Harmonizar a legislação de todos os Estados Membros da UE;
- Contribuir para um mercado único europeu de dados garantindo a livre circulação de dados pessoais entre os Estados da UE.

O RGPD vem definir um conjunto de princípios relativos à recolha e tratamento de dados pessoais. O RGPD contém um conjunto de disposições e requisitos relativos ao tratamento de dados pessoais dos indivíduos e impõe que os controladores de dados pessoais implementem medidas técnicas e organizacionais apropriadas para cumprir com os princípios de proteção de dados.

O regulamento exige que sejam implementadas medidas para garantir o princípio da responsabilidade, o qual exige que o responsável pelo tratamento dos dados se responsabilize pelo que faz com os dados pessoais.

Esse projeto utiliza em seus testes dados abertos, informa suas características e como podem ser encontrados, como foram obtidos e suas respectivas fontes.

6. Caracterização da máquina preditiva

Caracterização do algoritmo utilizado

O que é Análise de Cluster?

Pelo dicionário, a definição de “agrupar” é “localizar-se próximo, um em relação ao outro; encontrar-se”. Em análise de dados, agrupar é uma modalidade de aprendizado de máquina no qual o resultado vem da aplicação de modelos matemáticos e estatísticos.

A análise de cluster se enquadra nos métodos não supervisionado de aprendizagem automática.

Dentro de aprendizagem automática temos métodos de aprendizagem supervisionada e não supervisionada, além de outros. Na aprendizagem não supervisionada, não se especifica uma variável de destino para a máquina, em vez disso, perguntamos-lhe “O que me pode dizer sobre X?”. Mais especificamente, podemos fazer perguntas como, dado um enorme conjunto de dados X, “Quais são os cinco melhores grupos que podemos fazer de X?” ou “Que recursos ocorrem juntos com mais frequência no X?”.

Modelos não-supervisionados significam que a análise será exploratória, ou seja, serão explorados padrões, tendências ou comportamentos sem ideia do que o modelo pode resultar. Sendo exploratória, não há que se falar em inferência, mas apenas em diagnóstico. Nesses modelos, serão organizados e manipulados dados (observações e variáveis) sem uma variável que possa conduzir a máquina a encontrar um caminho previamente conhecido.

Ao analisar uma base de dados, um dos principais desafios do analista é resumir a informação coletada. Em muitos casos, quando contamos com um grande número de observações, pode ser de interesse criar grupos. Dentro de cada grupo os elementos devem ser semelhantes entre si e diferentes dos elementos dentro dos outros grupos.

A análise de clusters é uma técnica de análise multivariada que tem como principal objetivo o agrupamento de elementos. Este agrupamento é efetuado de forma que elementos pertencentes ao mesmo grupo tenham características semelhantes e elementos de diferentes grupos tenham características dissemelhantes. Genericamente, parte-se de um conjunto com n observações e pretende-se formar k grupos com um menor número de observações. Análise de cluster é amplamente aplicado em diversos casos relacionado a análise de dados, como:

Segmentação de clientes de características semelhantes

Identificação de anomalias, o que pode levar a possíveis identificações de fraudes

Modelos de recomendação otimizando a distribuição de produtos para determinados grupos de clientes.

Para criar categorias que não são tão lógicas, mas que fazem sentido do ponto de vista estatístico

Para definir a semelhança – ou diferença – entre os elementos é usada uma função de distância, que precisa ser definida considerando o contexto do problema em questão.

Para a construção desses grupos usam-se métodos hierárquicos ou métodos não hierárquicos. Nesse trabalho estou utilizando o método não hierárquico k-means.

Os métodos não-hierárquicos (particionados), foram desenvolvidos para agrupar elementos em k grupos, onde k é a quantidade de grupos definida previamente, assim, os métodos não-hierárquicos da análise de cluster são caracterizados pela necessidade de definir uma partição inicial e pela flexibilidade, uma vez que os elementos podem ser trocados de grupo durante a execução do algoritmo.

Nem todos os valores k apresentam grupos satisfatórios, sendo assim, aplica-se o método várias vezes para diferentes valores de k, escolhendo os resultados que apresentem melhor interpretação dos grupos ou uma melhor representação gráfica. A ideia central da maioria dos métodos por particionamento é escolher uma partição inicial dos elementos e, em seguida, alterar os membros dos grupos para obter-se a melhor partição. Nesse trabalho fazemos o agrupamento com os 5 possíveis números que estão dentro da escala do número mágico 7+-2, e escolhemos o que tem melhor pontuação.

Método K-means

O método K-means, trata-se de um método de Análise de Clusters não hierárquico muito utilizado principalmente por ser um método difundido na maioria dos softwares estatísticos. Esse método tem a particularidade de ser de fácil aplicação sobretudo quando a dimensão da amostra é grande. O método K-means parte de um número de grupos (clusters) definido a priori e calcula os pontos que representam os centros destes grupos e que são espalhados de forma homogénea no conjunto de respostas obtidas até alcançar um equilíbrio.

K-means e muitas vezes referido como algoritmo de Lloyd. Em termos básicos, o algoritmo tem três etapas. A primeira etapa escolhe os centroides iniciais, sendo o método mais básico a escolha de um ponto do conjunto de dados. Após a inicialização, o K-means faz um ciclo com as duas etapas seguintes. Na segunda etapa, para cada ponto, determina-se o seu centroide mais próximo e associa- se o ponto a esse cluster. Na terceira etapa cria novos centroides (ou recentra os centroides anteriores) considerando o valor médio de todos os pontos associados previamente a cada cluster.

A diferença entre o centroide antigo e o novo é computada e o algoritmo repete as duas últimas etapas até esse valor ser menor que um determinado limite. Noutras palavras, repete-se o ciclo até que os centroides se acabem por fixar definitivamente (ou, em alternativa, até se atingir um número máximo de iterações).

7.Caracterização dos datasets utilizados

Avaliações de hotéis (Hotels Reviews(booking.com))

Como foi obtido?

- https://www.kaggle.com/datasets

Informações do conjunto de dados:

- Este conjunto de dados contém avaliações de hotéis em Los Angeles do popular site de viagens booking.com. As avaliações são extraídas do site, adaptadas e organizadas em um arquivo csv.

Informações das variáveis:

- nome: O nome do hotel. (Corda)
- Pontuação geral: Pontuação geral do hotel, numa escala de 1 a 10. (Float)
- Limpeza: Pontuação de limpeza do hotel, numa escala de 1 a 10. (Float)
- Conforto: Pontuação de conforto do hotel, numa escala de 1 a 10. (Float)
- Instalações: Pontuação das instalações do hotel, numa escala de 1 a 10. (Float)
- Funcionários: Pontuação dos funcionários do hotel, em uma escala de 1 a 10. (Float)
- Custo-benefício: a pontuação de custo-benefício do hotel, em uma escala de 1 10. (Float)
- Wi-Fi gratuito: Se o hotel oferece ou não Wi-Fi gratuito, em uma escala de 1 a 10. (Float)
- Localização: pontuação da localização do hotel, em uma escala de 1 a 5. (Float)

Clientes_Shopping (Mall_Customers)

Como o conjunto de dados foi obtido?

- https://www.kaggle.com/datasets

Informações do conjunto de dados:

- Este arquivo contém as informações básicas (ID, idade, sexo, renda, pontuação de gastos) sobre os clientes, e é baseado em clientes de um determinado shopping. Há um total de 200 linhas e 5 colunas.

Informações das variáveis do conjunto de dados:

- CustomerID - ID exclusivo atribuído ao cliente
- Sexo - Sexo do cliente
- Idade - Idade do cliente
- Receita Anual (k$) - Receita Anual do cliente
- Pontuação de gastos (1-100) - Pontuação atribuída pelo shopping com base no comportamento do cliente e na natureza do gasto

Consumo_de_energia_doméstica (household_power_consumption)

Como foi obtido?

- https://archive.ics.uci.edu/ml/datasets

Fonte:

- Georges Hebrail, Pesquisador Sênior, EDF R&D, Clamart, França
- Alice Berard, Mestrado em Engenharia Estágio na EDF R&D, Clamart, França

Informações do conjunto de dados:

- Este arquivo contém 2075259 medições do consumo de energia elétrica coletadas numa residência em Sceaux (7 km de Paris, França) entre dezembro de 2006 e novembro de 2010 (47 meses).

Informações das variáveis:

- Date - Data no formato dd/mm/aaaa
- Time - hora no formato hh:mm:ss
- Global_active_power: potência ativa média global por minuto da residência (em kilowatt)
- Global_reactive_power: potência reativa média global por minuto da residência (em quilowatt)
- Tensão: tensão média por minuto (em volt)
- Global_intensity: intensidade de corrente média global por minuto doméstica (em ampere)

- Sub_metering_1: submedição de energia nº 1 (em watt-hora de energia ativa). Corresponde à cozinha, contendo maioritariamente uma máquina de lavar loiça, um forno e um micro-ondas (as placas eléctricas não são eléctricas mas sim a gás).
- Sub_metering_2: submedição de energia nº 2 (em watt-hora de energia ativa). Corresponde à lavanderia, contendo máquina de lavar, secadora, geladeira e luminária.
- Sub_metering_3: submedição de energia nº 3 (em watt-hora de energia ativa). Corresponde a um esquentador eléctrico e a um ar condicionado.

8.Análise e exploratória dos dados

Avaliações de hotéis (Hotels Reviews(booking.com))

Importando as bibliotecas (Figura 5).

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.metrics import silhouette_score
```

Figura 6 - Imports (booking.com)

Fonte: elaborado pelo autor.

Carregando os dados (Figura 6).

```
dataset = pd.read_csv('la_hotels_data.csv', delimiter=';')
```

Figura 7 - carreg (booking.com)

Fonte: elaborado pelo autor.

O dataset está no mesmo diretório do arquivo do programa que está sendo executado.

Exibindo as cinco primeiras linhas do conjunto de dados (Figura 7).

```
print(dataset.head())
```

	Nome	Pontuacao	Limpeza	Conforto	...	Funcionarios	Custo-beneficio	WiFi_Gratuito	Localicao
0	Sheraton Gateway Los Angeles	8.1	8.5	8.4	...	7.8	7.4	7.2	8.2
1	The Jeremy Hotel West Hollywood	8.9	9.3	9.0	...	8.9	8.2	9.5	9.1
2	Hampton Inn & Suites Santa Monica	8.3	8.8	8.6	...	8.2	7.2	8.4	8.9
3	Beverly Wilshire	7.0	8.9	9.1	...	8.7	9.2	7.6	8.4
4	The London West Hollywood at Beverly Hills	8.8	9.2	9.1	...	8.9	7.9	8.9	8.9

[5 rows x 9 columns]

Figura 8 - Head (booking.com)

Fonte: elaborado pelo autor.

As dimensões do conjunto de dados (Figura 8).

```
print(dataset.shape)
```

```
(582, 9)
```

Figura 9 - dim (booking.com)

Fonte: elaborado pelo autor.

Verificando os tipos de campos (Figura 9).

```
print(dataset.dtypes)
```

```
Nome                object
Pontuacao          float64
Limpeza            float64
Conforto           float64
Instalacoes        float64
Funcionarios       float64
Custo-beneficio    float64
WiFi_Gratuito      float64
Localicao          float64
dtype: object
```

Figura 10 - Tipo (booking.com)

Fonte: elaborado pelo autor.

Informações gerais do conjunto de dados (Figura 10).

```
print(dataset.info())
```

```
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Nome             582 non-null    object
 1   Pontuacao        582 non-null    float64
 2   Limpeza          582 non-null    float64
 3   Conforto         582 non-null    float64
 4   Instalacoes      582 non-null    float64
 5   Funcionarios     582 non-null    float64
 6   Custo-beneficio  582 non-null    float64
 7   WiFi_Gratuito    582 non-null    float64
 8   Localicao        582 non-null    float64
dtypes: float64(8), object(1)
memory usage: 41.0+ KB
```

Figura 11 - Info (booking.com)

Fonte: elaborado pelo autor.

Removendo a primeira coluna do conjunto de dados, ela não é necessária na análise (Figura 11).

```
hotels = dataset.iloc[0:, 1:]
```

Figura 12 - Iloc (booking.com)

Fonte: elaborado pelo autor.

Exibindo as cinco primeiras linhas do conjunto de dados, agora sem o nome dos hotéis (Figura 12).

```
print(hotels.head())
```

	Pontuacao	Limpeza	Conforto	Instalacoes	Funcionarios	Custo-beneficio	WiFi_Gratuito	Localicao
0	8.1	8.5	8.4	8.0	7.8	7.4	7.2	8.2
1	8.9	9.3	9.0	8.7	8.9	8.2	9.5	9.1
2	8.3	8.8	8.6	8.2	8.2	7.2	8.4	8.9
3	7.0	8.9	9.1	9.0	8.7	9.2	7.6	8.4
4	8.8	9.2	9.1	8.9	8.9	7.9	8.9	8.9

Figura 13 - Head (booking.com)

Fonte: elaborado pelo autor.

Obtenção dos valores de cada variável num formato de array (Figura 13).

```
v_hotels = hotels.values
```

```
[[8.1 8.5 8.4 ... 7.4 7.2 8.2]
 [8.9 9.3 9.  ... 8.2 9.5 9.1]
 [8.3 8.8 8.6 ... 7.2 8.4 8.9]
 ...
 [8.2 8.7 8.2 ... 7.4 9.2 8.5]
 [7.8 8.1 8.2 ... 7.  7.5 8.3]
 [8.2 8.7 8.1 ... 7.3 9.  8.9]]
```

Figura 14 - Valores (booking.com)

Fonte: elaborado pelo autor.

Verificando se há valores nulos e somando eles por coluna (Figura 14).

```
print(hotels.isnull().sum())
```

```
Pontuacao          0
Limpeza            0
Conforto           0
Instalacoes        0
Funcionarios       0
Custo-beneficio    0
WiFi_Gratuito      0
Localizacao        0
dtype: int64
```

Figura 15 - Isnull (booking.com)

Fonte: elaborado pelo autor.

Aplicando redução de dimensionalidade no array das variáveis (Figura 15).

```
pca = PCA(n_components=2).fit_transform(v_hotels)
```

```
[[ 0.26304378 -1.24861004]
 [-1.68377808  0.90920545]
 [-0.71090575 -0.67260173]
 ...
 [-0.35424602 -0.35478666]
 [ 0.52144347 -1.78020468]
 [-0.67089609 -0.6548762 ]]
```

Figura 16 – pca (booking.com)

Fonte: elaborado pelo autor.

Função que devolve a melhor quantidade de cluster considerando os valores que estão entre 7+-2 (Figura 16).

```
def calcular_melhor_k(x):

    modelo_v1 = KMeans(n_clusters=5)
    modelo_v1.fit_predict(x)
    labels = modelo_v1.labels_
    k5 = silhouette_score(x, labels, metric='euclidean')

    modelo_v2 = KMeans(n_clusters=6)
    modelo_v2.fit_predict(x)
    labels = modelo_v2.labels_
    k6 = silhouette_score(x, labels, metric='euclidean')

    modelo_v3 = KMeans(n_clusters=7)
    modelo_v3.fit_predict(x)
    labels = modelo_v3.labels_
    k7 = silhouette_score(x, labels, metric='euclidean')

    modelo_v4 = KMeans(n_clusters=8)
    modelo_v4.fit_predict(x)
    labels = modelo_v4.labels_
    k8 = silhouette_score(x, labels, metric='euclidean')

    modelo_v5 = KMeans(n_clusters=9)
    modelo_v5.fit_predict(x)
    labels = modelo_v5.labels_
    k9 = silhouette_score(x, labels, metric='euclidean')

    if (k5 > k6 and k5 > k7 and k5 > k8 and k5 > k9):
        modelo = modelo_v1
    elif (k6 > k5 and k6 > k7 and k6 > k8 and k6 > k9):
        modelo = modelo_v2
    elif (k7 > k5 and k7 > k6 and k7 > k8 and k7 > k9):
        modelo = modelo_v3
    elif (k8 > k5 and k8 > k6 and k8 > k7 and k8 > k9):
        modelo = modelo_v4
    else:
        modelo = modelo_v5

    return modelo
```

Figura 17 - Func7+-2 (booking.com)

Fonte: elaborado pelo autor.

Função que executa o aplicativo k-means e retorna a melhor quantidade de cluster dentre os valores do número mágico 7+-2. Para determinar a melhor quantidade cluster estou o silhouette_score. A pontuação de Silhueta é calculada usando a distância média intra-cluster e a distância média do cluster mais próximo para cada amostra. Trata-se da distância entre uma amostra e o cluster mais próximo do qual a amostra não faz parte; é um método de interpretação e validação de consistência dentro de agrupamentos de dados. A técnica fornece uma representação sucinta de quão bem cada objeto foi classificado.

Chamando a função (Figura 17).

```
modelo = calcular_melhor_k(pca)
```

```
KMeans(n_clusters=5)
```

Figura 18 – Execut func (booking.com)

Fonte: elaborado pelo autor.

Para esse conjunto de dados, o modelo com cinco cluster foi o de melhor pontuação.

Lista com os nomes das colunas (Figura 18).

```
names = ['Nome', 'Pontuacao', 'Limpeza', 'Conforto', 'Instalacoes',
         'Funcionarios', 'Custo-beneficio', 'WiFi_Gratuito', 'Localizacao']
```

Figura 19 - Names (booking.com)

Fonte: elaborado pelo autor.

Incluindo a variável com número do cluster na base de clientes (Figura 19).

```
cluster_map = pd.DataFrame(dataset, columns=names)
cluster_map['cluster'] = modelo.labels_
```

Figura 20 - Nova variável (booking.com)

Fonte: elaborado pelo autor.

Impressão das cinco primeiras e cinco últimas linhas do conjunto de dados com seu respetivo cluster (Figura 20).

	Nome	Pontuacao	Limpeza	Conforto	...	Custo-beneficio	WiFi_Gratuito	Localizacao	cluster
0	Sheraton Gateway Los Angeles	8.1	8.5	8.4	...	7.4	7.2	8.2	3
1	The Jeremy Hotel West Hollywood	8.9	9.3	9.0	...	8.2	9.5	9.1	0
2	Hampton Inn & Suites Santa Monica	8.3	8.8	8.6	...	7.2	8.4	8.9	2
3	Beverly Wilshire	7.0	8.9	9.1	...	9.2	7.6	8.4	2
4	The London West Hollywood at Beverly Hills	8.8	9.2	9.1	...	7.9	8.9	8.9	0
..	...	...	...	...	...	...	...	...	...
577	Downtown Mansion Hostel	8.8	8.5	8.8	...	8.8	10.0	8.6	0
578	Royal Pagoda Motel	7.6	7.7	7.4	...	7.3	7.8	7.9	3
579	Bel Air Villa	8.2	8.7	8.2	...	7.4	9.2	8.5	2
580	Pelham Apartment 8	7.8	8.1	8.2	...	7.0	7.5	8.3	3
581	Palihotel Melrose	8.2	8.7	8.1	...	7.3	9.0	8.9	2

[582 rows x 10 columns]

Figura 21 - Novo dataset (booking.com)

Fonte: elaborado pelo autor.

Salvando o novo dataset, agora com uma nova variável com o número de cluster (Figura 21).

```
cluster_map.to_csv('hotels_clusters.csv')
```

Figura 22 – novoHotel

Fonte: elaborado pelo autor.

Calcula média de cluster por pontuação (Figura 22).

```
print(cluster_map.groupby('cluster')['Pontuacao'].mean())
```

```
cluster
0    8.935404
1    7.666667
2    7.602679
3    8.252838
4    8.767925
Name: Pontuacao, dtype: float64
```

Figura 23 – Med/pont (booking.com)

Fonte: elaborado pelo autor.

Calcula média de cluster por conforto (Figura 23).

```
print(cluster_map.groupby('cluster')['Conforto'].mean())
```

```
cluster
0    9.124224
1    7.866667
2    7.658929
3    8.398690
4    8.915094
Name: Conforto, dtype: float64
```

Figura 24 - Med/conf (booking.com)

Fonte: elaborado pelo autor.

Calcula média de cluster por custo-benefício (Figura 24).

```
print(cluster_map.groupby('cluster')['Custo-beneficio'].mean())

cluster
0    8.538509
1    7.303704
2    7.057143
3    7.733624
4    8.341509
Name: Custo-beneficio, dtype: float64
```

Figura 25 - Méd/cb (booking.com)

Fonte: elaborado pelo autor.

Quantidade de hotéis em cada grupo (Figura 25).

```
print(cluster_map.groupby('cluster').count())

         cluster
cluster
0            161
1             27
2            112
3            229
4             53
```

Figura 26 - count (booking.com)

Fonte: elaborado pelo autor.

Visualização os grupos em gráfico de dispersão (Figura 26).

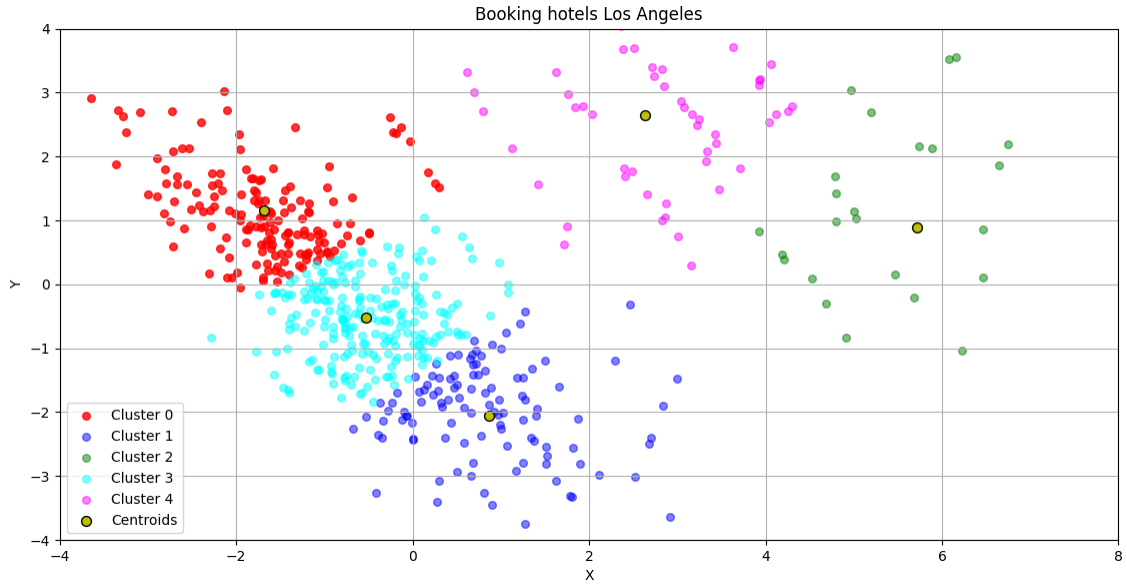


Figura 27 – Dispersão (booking.com)

Fonte: elaborado pelo autor.

Clientes_Shopping (Mall_Customers)

Importando as bibliotecas (Figura 27).

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
```

Figura 28 - Imports (Mall_Customers)

Fonte: elaborado pelo autor.

Carregando os dados (Figura 28).

```
dataset = pd.read_csv('Mall_Customers.csv', delimiter=',')
```

Figura 29 - Carreg (Mall_Customers)

Fonte: elaborado pelo autor.

O dataset está no mesmo diretório do arquivo do programa que está sendo executado.

Exibindo as cinco primeiras linhas do conjunto de dados (Figura 29).

```
dataset.head()
```

	IDCliente	Genero	Idade	Renda Anual (k$)	Pontuacao de Gastos (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

Figura 30 - Head (Mall_Customers)

Fonte: elaborado pelo autor.

As dimensões do conjunto de dados (Figura 30).

```
dataset.shape
```

```
(200, 5)
```

Figura 31 - dim (Mall_Customers)

Fonte: elaborado pelo autor.

Verificando os tipos de campos (Figura 31).

```
dataset.dtypes
```

```
IDCliente                    int64
Genero                      object
Idade                        int64
Renda Anual (k$)             int64
Pontuacao de Gastos (1-100)  int64
dtype: object
```

Figura 32 - Tipos (Mall_Customers)

Fonte: elaborado pelo autor.

Informações gerais do conjunto de dados (Figura 32).

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   IDCliente                   200 non-null    int64
 1   Genero                      200 non-null    object
 2   Idade                       200 non-null    int64
 3   Renda Anual (k$)            200 non-null    int64
 4   Pontuacao de Gastos (1-100)  200 non-null    int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

Figura 33 - Info (Mall_Customers)

Fonte: elaborado pelo autor.

Descrições estatísticas básicas gerais do Dataset (Figura 33).

```
dataset.describe()
```

	IDCliente	Idade	Renda Anual (k$)	Pontuacao de Gastos (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

Figura 34 - Desc (Mall_Customers)

Fonte: elaborado pelo autor.

Removendo as primeiras colunas do conjunto de dados, não são utilizadas nessa análise (Figura 34).

```
clientes = dataset.iloc[0:, [3, 4]]
```

Figura 35 - Iloc (Mall_Customers)

Fonte: elaborado pelo autor.

Visualizando como ficou o conjunto de dados (Figura 35).

```
clientes
```

	Renda Anual (k$)	Pontuacao de Gastos (1-100)
0	15	39
1	15	81
2	16	6
3	16	77
4	17	40
...	...	...
195	120	79
196	126	28
197	126	74
198	137	18
199	137	83

200 rows × 2 columns

Figura 36 - Visual (Mall_Customers)

Fonte: elaborado pelo autor.

Obtenção dos valores de cada variável num formato de array (Figura 36).

```
vclientes = clientes.values
```

Figura 37 - Values (Mall_Customers)

Fonte: elaborado pelo autor.

Verificando se há valores missing (Figura 37).

```
dataset.isnull().values.any()
```

```
False
```

Figura 38 - Missing (Mall_Customers)

Fonte: elaborado pelo autor.

Função que devolve a melhor quantidade de cluster considerando os valores que estão entre 7+-2 (Figura 38).

```
def calcular_melhor_k(x):

    modelo_v1 = KMeans(n_clusters=5)
    modelo_v1.fit_predict(x)
    labels = modelo_v1.labels_
    k5 = silhouette_score(x, labels, metric='euclidean')

    modelo_v2 = KMeans(n_clusters=6)
    modelo_v2.fit_predict(x)
    labels = modelo_v2.labels_
    k6 = silhouette_score(x, labels, metric='euclidean')

    modelo_v3 = KMeans(n_clusters=7)
    modelo_v3.fit_predict(x)
    labels = modelo_v3.labels_
    k7 = silhouette_score(x, labels, metric='euclidean')

    modelo_v4 = KMeans(n_clusters=8)
    modelo_v4.fit_predict(x)
    labels = modelo_v4.labels_
    k8 = silhouette_score(x, labels, metric='euclidean')

    modelo_v5 = KMeans(n_clusters=9)
    modelo_v5.fit_predict(x)
    labels = modelo_v5.labels_
    k9 = silhouette_score(x, labels, metric='euclidean')

    if (k5 > k6 and k5 > k7 and k5 > k8 and k5 > k9):
        modelo = modelo_v1
    elif (k6 > k5 and k6 > k7 and k6 > k8 and k6 > k9):
        modelo = modelo_v2
    elif (k7 > k5 and k7 > k6 and k7 > k8 and k7 > k9):
        modelo = modelo_v3
    elif (k8 > k5 and k8 > k6 and k8 > k7 and k8 > k9):
        modelo = modelo_v4
    else:
        modelo = modelo_v5

    return modelo
```

Figura 39 – Func7+-2 (Mall_Customers)

Fonte: elaborado pelo autor.

Função que executa o aplicativo k-means e retorna a melhor quantidade de cluster dentre os valores do número mágico 7+-2. Para determinar a melhor quantidade cluster estou o silhouette_score.

A pontuação de Silhueta é calculada usando a distância média intra-cluster e a distância média do cluster mais próximo para cada amostra. Trata-se da distância entre uma amostra e o cluster mais próximo do qual a amostra não faz parte; é um método de interpretação e validação de consistência dentro de agrupamentos de dados. A técnica fornece uma representação sucinta de quão bem cada objeto foi classificado.

Chamando a função (Figura 39).

```
modelo = calcular_melhor_k(vclientes)
modelo
```

```
KMeans(n_clusters=5)
```

Figura 40 - Exec função (Mall_Customers)

Fonte: elaborado pelo autor.

Para esse conjunto de dados, o modelo com cinco cluster foi o de melhor pontuação.

Lista com os nomes das colunas (Figura 40).

```
names = ['IDCliente', 'Genero', 'Idade',
         'Renda Anual (k$)', 'Pontuacao de Gastos (1-100)']
```

Figura 41 - Names (Mall_Customers)

Fonte: elaborado pelo autor.

Incluindo a variável com número do cluster na base de clientes e mostrando as primeiras e últimas linhas do conjunto de dados com seu respetivo cluster (Figura 41).

```
cluster_map = pd.DataFrame(dataset, columns=names)
cluster_map['cluster'] = modelo.labels_
cluster_map
```

	IDCliente	Genero	Idade	Renda Anual (k$)	Pontuacao de Gastos (1-100)	cluster
0	1	Male	19	15	39	4
1	2	Male	21	15	81	1
2	3	Female	20	16	6	4
3	4	Female	23	16	77	1
4	5	Female	31	17	40	4
...	...	...	...	...	...	...
195	196	Female	35	120	79	3
196	197	Female	45	126	28	0
197	198	Male	32	126	74	3
198	199	Male	32	137	18	0
199	200	Male	30	137	83	3

200 rows × 6 columns

Figura 42 - Cluster_map (Mall_Customers)

Fonte: elaborado pelo autor.

Salvando novo dataset, agora com número de cluster incluído (Figura 42).

```
cluster_map.to_csv('novoClientes.csv')
```

Figura 43 - Novo Dataset (Mall_Customers)

Fonte: elaborado pelo autor.

Separando cada cluster (Figura 43).

```
clus0 = cluster_map.loc[cluster_map.cluster==0].cluster
clus1 = cluster_map.loc[cluster_map.cluster==1].cluster
clus2 = cluster_map.loc[cluster_map.cluster==2].cluster
clus3 = cluster_map.loc[cluster_map.cluster==3].cluster
clus4 = cluster_map.loc[cluster_map.cluster==4].cluster
```

Figura 44 - Sep cluster (Mall_Customers)

Fonte: elaborado pelo autor.

Calcula média de idade de cada cluster (Figura 44).

```
cluster_map.groupby('cluster')['Idade'].mean()
```

```
cluster
0    41.114286
1    25.272727
2    42.716049
3    32.692308
4    45.217391
Name: Idade, dtype: float64
```

Figura 45 - Med/idade (Mall_Customers)

Fonte: elaborado pelo autor.

Calcula média de renda anual por cluster (Figura 45).

```
cluster_map.groupby('cluster')['Renda Anual (k$)'].mean()
```

```
cluster
0    88.200000
1    25.727273
2    55.296296
3    86.538462
4    26.304348
Name: Renda Anual (k$), dtype: float64
```

Figura 46 - Med/renda (Mall_Customers)

Fonte: elaborado pelo autor.

Calcula média de pontuação por cluster (Figura 46).

```
cluster_map.groupby('cluster')['Pontuacao de Gastos (1-100)'].mean()
```

```
cluster
0    17.114286
1    79.363636
2    49.518519
3    82.128205
4    20.913043
Name: Pontuacao de Gastos (1-100), dtype: float64
```

Figura 47 - Med/pont (Mall_Customers)

Fonte: elaborado pelo autor.

Quantidade de clientes em cada grupo (Figura 47).

```
cluster_map.groupby('cluster')['IDCliente'].count()
```

```
cluster
0    35
1    22
2    81
3    39
4    23
Name: IDCliente, dtype: int64
```

Figura 48 - Count (Mall_Customers)

Fonte: elaborado pelo autor.

Percentual de clientes por cluster (Figura 48).

```
g = (cluster_map.groupby('cluster')['IDCliente'].count())
s = g.sum()
c0, c1, c2, c3, c4 = clus0.count(), clus1.count(), clus2.count(), clus3.count(), clus4.count()
pct = [c0/s, c1/s, c2/s, c3/s, c4/s]
print('O cluster 0 tem %.2f dos registros'  %(pct[0]))
print('O cluster 1 tem %.2f dos registros'  %(pct[1]))
print('O cluster 2 tem %.2f dos registros'  %(pct[2]))
print('O cluster 3 tem %.2f dos registros'  %(pct[3]))
print('O cluster 4 tem %.2f dos registros'  %(pct[4]))
```

```
O cluster 0 tem 0.17 dos registros
O cluster 1 tem 0.11 dos registros
O cluster 2 tem 0.41 dos registros
O cluster 3 tem 0.20 dos registros
O cluster 4 tem 0.12 dos registros
```

Figura 49 - Pct/cluster (Mall_Customers)

Fonte: elaborado pelo autor.

Visualização os grupos em gráfico de dispersão (Figura 49 e 50).

```
clus = modelo.labels_
plt.figure(figsize=(18,8))
plt.scatter(vclientes[clus == 0, 0], vclientes[clus == 0, 1], s=60, c='r', label='Cluster 0', alpha=0.7)
plt.scatter(vclientes[clus == 1, 0], vclientes[clus == 1, 1], s=60, c='b', label='Cluster 1', alpha=0.7)
plt.scatter(vclientes[clus == 2, 0], vclientes[clus == 2, 1], s=60, c='g', label='Cluster 2', alpha=0.7)
plt.scatter(vclientes[clus == 3, 0], vclientes[clus == 3, 1], s=60, c='c', label='Cluster 3', alpha=0.7)
plt.scatter(vclientes[clus == 4, 0], vclientes[clus == 4, 1], s=60, c='m', label='Cluster 4', alpha=0.7)
plt.scatter(modelo.cluster_centers_[:, 0], modelo.cluster_centers_[:, 1], s=120, c='y', edgecolor="black", label='Centroids')
plt.title("Modelo do conjunto clientes do shopping")
plt.xlabel('Renda Anual (k$)')
plt.ylabel('Pontuacao de Gastos (1-100)')
plt.legend()
plt.grid(True)
plt.show()
```

Figura 50 - CodDisp (Mall_Customers)

Fonte: elaborado pelo autor.

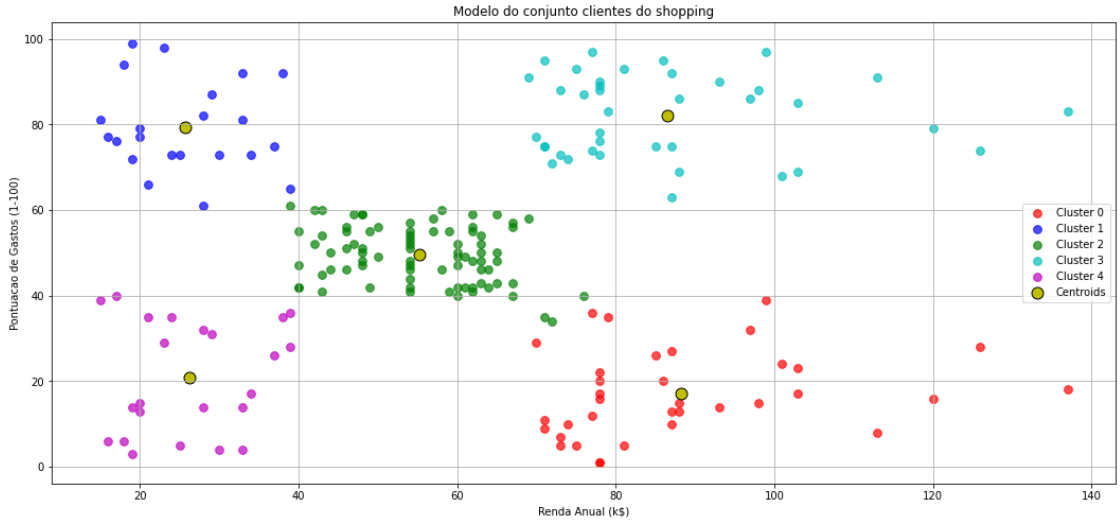


Figura 51 - GráfDisp (Mall_Customers)

Fonte: elaborado pelo autor.

Consumo_de_energia_doméstica (household_power_consumption)

Importando as bibliotecas (Figura 51).

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.model_selection import train_test_split
from scipy.spatial.distance import cdist, pdist
from sklearn.metrics import silhouette_score
```

Figura 52 - Imports (HPC)

Fonte: elaborado pelo autor.

Carregando os dados (Figura 52).

```
dataset = pd.read_csv('household_power_consumption.txt', delimiter = ';', low_memory = False)
```

Figura 53 - read (HPC)

Fonte: elaborado pelo autor.

O dataset está no mesmo diretório do arquivo do programa que está sendo executado.

Exibindo as cinco primeiras linhas do conjunto de dados (Figura 53).

```
dataset.head()
```

	Date	Time	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
0	16/12/2006	17:24:00	4.216	0.418	234.840	18.400	0.000	1.000	17.0
1	16/12/2006	17:25:00	5.360	0.436	233.630	23.000	0.000	1.000	16.0
2	16/12/2006	17:26:00	5.374	0.498	233.290	23.000	0.000	2.000	17.0
3	16/12/2006	17:27:00	5.388	0.502	233.740	23.000	0.000	1.000	17.0
4	16/12/2006	17:28:00	3.666	0.528	235.680	15.800	0.000	1.000	17.0

Figura 54 - head (HPC)

Fonte: elaborado pelo autor.

As dimensões do conjunto de dados (Figura 54).

```
dataset.shape
```

```
(2075259, 9)
```

Figura 55 - Dim (HPC)

Fonte: elaborado pelo autor.

Verificando os tipos de campos (Figura 55).

```
dataset.dtypes
```

```
Date                    object
Time                    object
Global_active_power     object
Global_reactive_power   object
Voltage                 object
Global_intensity        object
Sub_metering_1          object
Sub_metering_2          object
Sub_metering_3         float64
dtype: object
```

Figura 56 - Tipos (HPC)

Fonte: elaborado pelo autor.

Informações gerais do conjunto de dados (Figura 56).

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075259 entries, 0 to 2075258
Data columns (total 9 columns):
 #   Column                 Dtype  
---  ------                 -----  
 0   Date                   object 
 1   Time                   object 
 2   Global_active_power    object 
 3   Global_reactive_power  object 
 4   Voltage                object 
 5   Global_intensity       object 
 6   Sub_metering_1         object 
 7   Sub_metering_2         object 
 8   Sub_metering_3         float64
dtypes: float64(1), object(8)
memory usage: 142.5+ MB
```

Figura 57 - Info (HPC)

Fonte: elaborado pelo autor.

Verificando se há valores missing (Figura 57).

```
dataset.isnull().values.any()
```

```
True
```

Figura 58 - Missing (HPC)

Fonte: elaborado pelo autor.

Verificando onde estão os valores missing (Figura 58).

```
dataset.isnull().sum()
```

```
Date                       0
Time                       0
Global_active_power        0
Global_reactive_power      0
Voltage                    0
Global_intensity           0
Sub_metering_1             0
Sub_metering_2             0
Sub_metering_3         25979
dtype: int64
```

Figura 59 - Sum (HPC)

Fonte: elaborado pelo autor.

Remove os registros com valores nulos e remove as dias primeiras colunas do conjunto de dados, não são utilizadas nessa análise (Figura 59).

```
df = dataset.iloc[0:, 2:9].dropna()
```

Figura 60 - iloc (HPC)

Fonte: elaborado pelo autor.

Verificando os valores nulos e visualizando como ficou o conjunto de dados (Figura 60 e 61).

```
df.isnull().values.any()
```

```
False
```

Figura 61 - any (HPC)

Fonte: elaborado pelo autor.

```
df.isnull().sum()
```

```
Global_active_power      0
Global_reactive_power    0
Voltage                  0
Global_intensity         0
Sub_metering_1           0
Sub_metering_2           0
Sub_metering_3           0
dtype: int64
```

Figura 62 - Sum2 (HPC)

Fonte: elaborado pelo autor.

Visualização do Data Frame (Figura 62).

```
df
```

	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
0	4.216	0.418	234.840	18.400	0.000	1.000	17.0
1	5.360	0.436	233.630	23.000	0.000	1.000	16.0
2	5.374	0.498	233.290	23.000	0.000	2.000	17.0
3	5.388	0.502	233.740	23.000	0.000	1.000	17.0
4	3.666	0.528	235.680	15.800	0.000	1.000	17.0
...	...	...	...	...	...	...	...
2075254	0.946	0.000	240.430	4.000	0.000	0.000	0.0
2075255	0.944	0.000	240.000	4.000	0.000	0.000	0.0
2075256	0.938	0.000	239.820	3.800	0.000	0.000	0.0
2075257	0.934	0.000	239.700	3.800	0.000	0.000	0.0
2075258	0.932	0.000	239.550	3.800	0.000	0.000	0.0

2049280 rows × 7 columns

Figura 63 - Df (HPC)

Fonte: elaborado pelo autor.

Obtenção dos valores de cada variável num formato de array (Figura 63).

```
vdf = df.values
```

```
vdf
```

```
array([['4.216', '0.418', '234.840', ..., '0.000', '1.000', 17.0],
       ['5.360', '0.436', '233.630', ..., '0.000', '1.000', 16.0],
       ['5.374', '0.498', '233.290', ..., '0.000', '2.000', 17.0],
       ...,
       ['0.938', '0.000', '239.820', ..., '0.000', '0.000', 0.0],
       ['0.934', '0.000', '239.700', ..., '0.000', '0.000', 0.0],
       ['0.932', '0.000', '239.550', ..., '0.000', '0.000', 0.0]],
      dtype=object)
```

Figura 64 - Values (HPC)

Fonte: elaborado pelo autor.

Vou utilizar uma amostra de apenas 1% dos dados para análise estudantil, isso por con ta da memória do computador (Figura 64).

```
df1, df2 = train_test_split(vdf, train_size = .01)
```

Figura 65 - Amostra (HPC)

Fonte: elaborado pelo autor.

O train-test-split é utilizado para dividir (e opcionalmente subamostrar) arrays ou matrizes em subconjuntos aleatórios.

Verificando as dimensões do conjunto de dados (Figura 65).

```
df1.shape
```

```
(20492, 7)
```

Figura 66 - Shape (HPC)

Fonte: elaborado pelo autor.

Aplicando redução de dimensionalidade no array das variáveis pegando os component es principais para análise e mostrando o percentual de razão das variáveis (Figura 66).

```
pca = PCA(n_components = 2).fit_transform(df1)
xpca = PCA(n_components = 2).fit(df1)
```

```
xpca.explained_variance_ratio_
```

```
array([0.49101153, 0.24081328])
```

Figura 67 - pca (HPC)

Fonte: elaborado pelo autor.

Função que devolve a melhor quantidade de cluster considerando os valores que estão entre 7+-2 (Figura 67).

```
def calcular_melhor_k(x):

    modelo_v1 = KMeans(n_clusters=5)
    modelo_v1.fit_predict(x)
    labels = modelo_v1.labels_
    k5 = silhouette_score(x, labels, metric='euclidean')

    modelo_v2 = KMeans(n_clusters=6)
    modelo_v2.fit_predict(x)
    labels = modelo_v2.labels_
    k6 = silhouette_score(x, labels, metric='euclidean')

    modelo_v3 = KMeans(n_clusters=7)
    modelo_v3.fit_predict(x)
    labels = modelo_v3.labels_
    k7 = silhouette_score(x, labels, metric='euclidean')

    modelo_v4 = KMeans(n_clusters=8)
    modelo_v4.fit_predict(x)
    labels = modelo_v4.labels_
    k8 = silhouette_score(x, labels, metric='euclidean')

    modelo_v5 = KMeans(n_clusters=9)
    modelo_v5.fit_predict(x)
    labels = modelo_v5.labels_
    k9 = silhouette_score(x, labels, metric='euclidean')

    if (k5 > k6 and k5 > k7 and k5 > k8 and k5 > k9):
        modelo = modelo_v1
    elif (k6 > k5 and k6 > k7 and k6 > k8 and k6 > k9):
        modelo = modelo_v2
    elif (k7 > k5 and k7 > k6 and k7 > k8 and k7 > k9):
        modelo = modelo_v3
    elif (k8 > k5 and k8 > k6 and k8 > k7 and k8 > k9):
        modelo = modelo_v4
    else:
        modelo = modelo_v5

    return modelo
```

Figura 68 - Funcao 7+-2 (HPC)

Fonte: elaborado pelo autor.

Função que executa o aplicativo k-means e retorna a melhor quantidade de cluster dentre os valores do número mágico 7+-2. Para determinar a melhor quantidade cluster estou o utilizando a biblioteca silhouette_score.

A pontuação de Silhueta é calculada usando a distância média intra-cluster e a distância média do cluster mais próximo para cada amostra. Trata-se da distância entre uma amostra e o cluster mais próximo do qual a amostra não faz parte; é um método de interpretação e validação de consistência dentro de agrupamentos de dados. A técnica fornece uma representação sucinta de quão bem cada objeto foi classificado.

Chamando a função (Figura 68).

```
modelo = calcular_melhor_k(pca)
modelo
```

```
KMeans(n_clusters=5)
```

Figura 69 - ExecFunc (HPC)

Fonte: elaborado pelo autor.

Para esse conjunto de dados, o modelo com cinco cluster foi o de melhor pontuação.

Lista com os nomes das colunas (Figura 69).

```
names = ['Global_active_power', 'Global_reactive_power', 'Voltage', 'Global_intensity',
         'Sub_metering_1', 'Sub_metering_2', 'Sub_metering_3']
```

Figura 70 - Names (HPC)

Fonte: elaborado pelo autor.

Incluindo a variável com número do cluster na base de clientes e mostrando as primeiras e últimas linhas do conjunto de dados com seu respetivo cluster (Figura 70).

```
cluster_map = pd.DataFrame(df1, columns=names)
cluster_map['cluster'] = modelo.labels_
cluster_map
```

	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3	cluster
0	1.506	0.184	238.910	6.200	0.000	0.000	18.0	1
1	0.518	0.362	243.870	2.600	0.000	1.000	1.0	0
2	0.332	0.274	244.070	1.600	0.000	2.000	1.0	0
3	2.978	0.128	238.860	12.400	0.000	0.000	18.0	1
4	0.258	0.000	246.080	1.000	0.000	0.000	0.0	0
...	...	...	...	...	...	...	...	...
20487	1.358	0.000	245.790	5.400	0.000	0.000	18.0	1
20488	2.396	0.052	237.000	10.000	0.000	0.000	17.0	1
20489	1.444	0.198	236.920	6.000	2.000	0.000	17.0	1
20490	1.288	0.056	236.890	5.400	0.000	0.000	17.0	1
20491	0.292	0.080	244.440	1.200	0.000	0.000	0.0	0

20492 rows × 8 columns

Figura 71 - ClusterMap (HPC)

Fonte: elaborado pelo autor.

Salvando novo dataset, agora com número de cluster incluído (Figura 71).

```
cluster_map.to_csv('novoConsEnergEletrica.csv')
```

Figura 72 - NovoDataset (HPC)

Fonte: elaborado pelo autor.

Separando cada cluster (Figura 72).

```
clus0 = cluster_map.loc[cluster_map.cluster==0].cluster
clus1 = cluster_map.loc[cluster_map.cluster==1].cluster
clus2 = cluster_map.loc[cluster_map.cluster==2].cluster
clus3 = cluster_map.loc[cluster_map.cluster==3].cluster
clus4 = cluster_map.loc[cluster_map.cluster==4].cluster
```

Figura 73 - SepCluster (HPC)

Fonte: elaborado pelo autor.

Transforma os tipos de dados em numérico para calcular a média de outras variáveis (u ma por vez) por cluster (Figura 73).

```
cluster_map['Global_active_power'] = pd.to_numeric(cluster_map['Global_active_power'])
cluster_map.groupby('cluster')['Global_active_power'].mean()
```

```
cluster
0    0.518791
1    1.789009
2    4.086525
3    6.075975
4    3.017139
Name: Global_active_power, dtype: float64
```

```
cluster_map.groupby('cluster')['Sub_metering_3'].mean()
```

```
cluster
0     0.416352
1    17.861622
2    17.633663
3    12.860759
4     0.403270
Name: Sub_metering_3, dtype: float64
```

Figura 74 - media (HPC)

Fonte: elaborado pelo autor.

Quantidade de registros por grupo (Figura 74).

```
cluster_map.groupby('cluster')['Sub_metering_3'].count()
```

```
cluster
0    12965
1     6475
2      606
3       79
4      367
Name: Sub_metering_3, dtype: int64
```

Figura 75 - Count (HPC)

Fonte: elaborado pelo autor.

Percentual de registros por cluster (Figura 75).

```
g = (cluster_map.groupby('cluster')['Sub_metering_3'].count())
s = g.sum()
c0, c1, c2, c3, c4 = clus0.count(), clus1.count(), clus2.count(), clus3.count(), clus4.count()
pct = [c0/s, c1/s, c2/s, c3/s, c4/s]
# pct
print('O cluster 0 tem %.2f dos registros'  %(pct[0]))
print('O cluster 1 tem %.2f dos registros'  %(pct[1]))
print('O cluster 2 tem %.2f dos registros'  %(pct[2]))
print('O cluster 3 tem %.2f dos registros'  %(pct[3]))
print('O cluster 4 tem %.3f dos registros'  %(pct[4]))
```

```
O cluster 0 tem 0.63 dos registros
O cluster 1 tem 0.31 dos registros
O cluster 2 tem 0.03 dos registros
O cluster 3 tem 0.02 dos registros
O cluster 4 tem 0.002 dos registros
```

Figura 76 - Pct (HPC)

Fonte: elaborado pelo autor.

Visualização dos grupos em gráfico de dispersão (Figura 76 e 77).

```
clus = modelo.labels_
plt.figure(figsize=(18,8))
plt.scatter(pca[clus == 0, 0], pca[clus == 0, 1], s=60, c='r', label='Cluster 0', alpha=0.5)
plt.scatter(pca[clus == 1, 0], pca[clus == 1, 1], s=60, c='b', label='Cluster 1', alpha=0.5)
plt.scatter(pca[clus == 2, 0], pca[clus == 2, 1], s=60, c='g', label='Cluster 2', alpha=0.5)
plt.scatter(pca[clus == 3, 0], pca[clus == 3, 1], s=60, c='c', label='Cluster 3', alpha=0.5)
plt.scatter(pca[clus == 4, 0], pca[clus == 4, 1], s=60, c='m', label='Cluster 4', alpha=0.5)
plt.scatter(modelo.cluster_centers_[:, 0], modelo.cluster_centers_[:, 1], s=120, c='y', edgecolor="black", label='Centroids')
plt.title("Modelo do conjunto cosumo de energia elétrica")
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()
plt.grid(True)
plt.show()
```

Figura 77 - CodDisp (HPC)

Fonte: elaborado pelo autor.

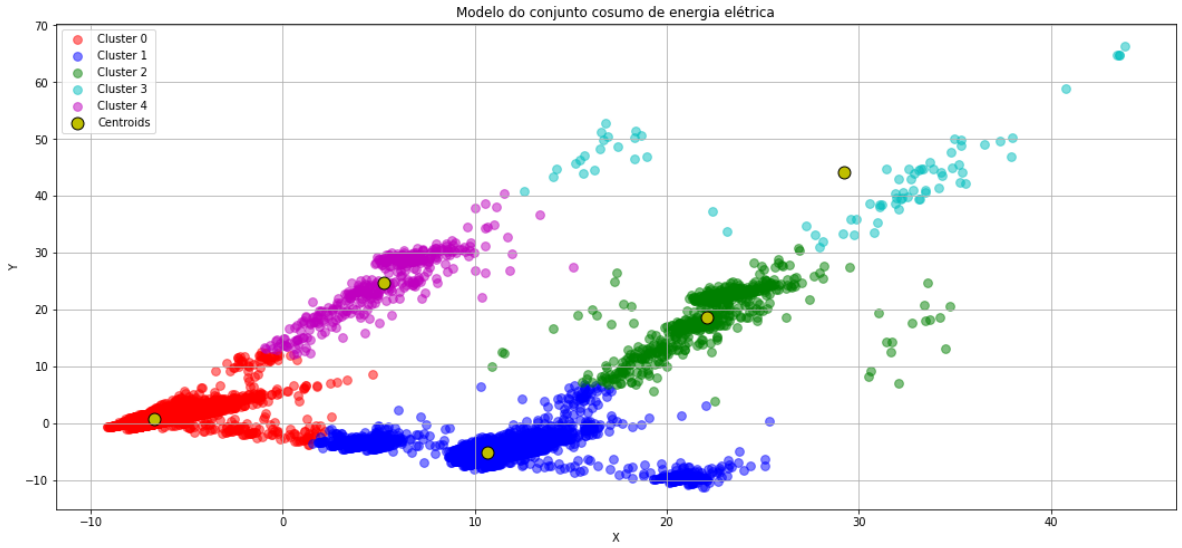


Figura 78 – GrafDisp (HPC)

Fonte: elaborado pelo autor.

9.Discussão de resultados

Como descobrir o número ideal de clusters?

A Curva de Cotovelo é uma outra forma muito utilizada de verificar o número ideal de cluster, sigo agora falando e mostrando o funcionamento desse método para um dos dataset's utilizados no projeto, mostrando que de fato o número ideal de cluster pra esse dataset' é mesmo 5.

A Inertia corresponde ao somatório dos erros quadráticos das instâncias de cada cluster.

Assim:

Mede o quanto os clusters estão separados entre eles

Mede a distância de cada dado para o centroid do seu cluster

Aplicamos fit() na inertia_ em busca de minimizar a inertia na escolha dos clusters

Quanto mais próximos entre si e o centroid, menor a inertia

A Curva de Cotovelo ou Método Elbow Curve é uma técnica usada para encontrar a quantidade ideal de clusters K. Este método testa a variância dos dados em relação ao número de clusters. O valor ideal de K é aquele que tem um menor Within Sum of Squares (WSS) e ao mesmo tempo o menor número de clusters. Chamamos de curva de cotovelo, porque a partir do ponto que seria o "cotovelo" não existe uma discrepância tão significativa em termos de variância. Dessa forma, a melhor quantidade de clusters K seria exatamente onde o cotovelo estaria.

Curva de Cotovelo/Método Elbow Curve (Consumo_de_energia_doméstica) (Figura 78 e 79).

```
wcss = []
for k in range(2,10):
  kmeans = KMeans(n_clusters=k)
  kmeans.fit(pca)
  wcss.append(kmeans.inertia_)

plt.figure(figsize=(18,8))
plt.plot(range(2, 10), wcss, '-rx')
plt.title('Curva de Cotovelo')
plt.xlabel('Numero de Clusters')
plt.ylabel('WCSS') #within cluster sum of squares
plt.legend()
plt.grid(True)
```

Figura 79 - CodElbow (HPC)

Fonte: elaborado pelo autor.

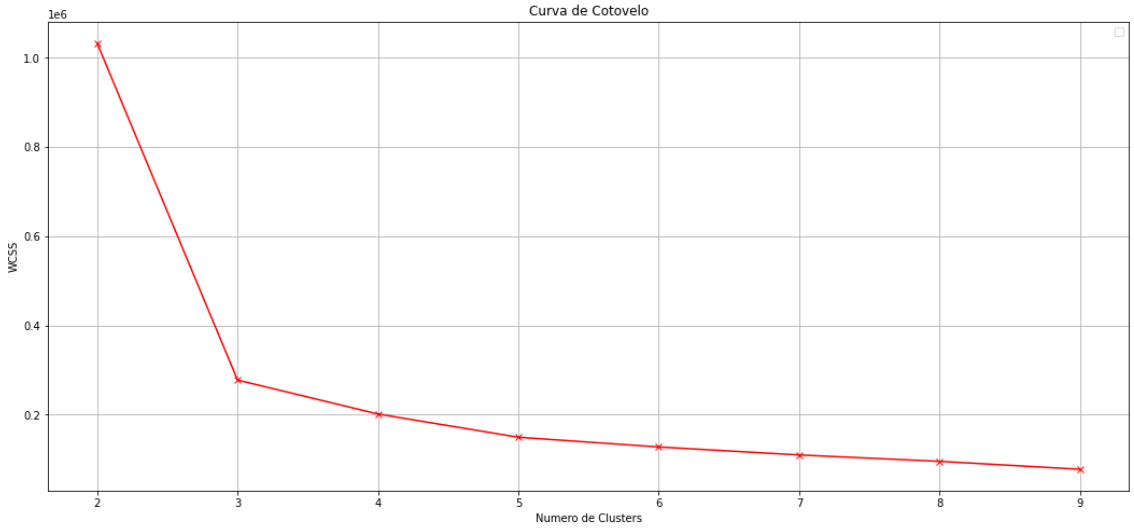


Figura 80 - GrafElbow (HPC)

Fonte: elaborado pelo autor.

Outro ponto importante para trazer é a possibilidade de fazer a redução de dimensionalidade para mais de duas dimensões, isso traz um melhor aproveitamento do conjunto de dados. Quando se pretende fazer uma análise estatística, provavelmente seja o melhor, mas quando se deseja fazer visualização em mapa, pode ser um pouco mais dificultosa a interpretação, mas também há a possibilidade para um olhar humano, mais ainda há a possibilidade mais a frente temos as imagens com as duas possibilidades de gráficos de dispersão para comparação.

Aplicando redução de dimensionalidade para 3 componentes principais

Consumo_de_energia_doméstica

Aplicando a redução de dimensionalidade para três variáveis (Figura 80).

```
pca = PCA(n_components = 3).fit_transform(df1)
xpca = PCA(n_components = 3).fit(df1)
```

```
xpca.explained_variance_ratio_
```

```
array([0.48694308, 0.24310109, 0.19393548])
```

Figura 81 - Pca (anexo)

Fonte: elaborado pelo autor.

Separando cada cluster (Figura 81).

```
clus0 = cluster_map.loc[cluster_map.cluster==0].cluster
clus1 = cluster_map.loc[cluster_map.cluster==1].cluster
clus2 = cluster_map.loc[cluster_map.cluster==2].cluster
clus3 = cluster_map.loc[cluster_map.cluster==3].cluster
clus4 = cluster_map.loc[cluster_map.cluster==4].cluster
clus5 = cluster_map.loc[cluster_map.cluster==5].cluster
clus6 = cluster_map.loc[cluster_map.cluster==6].cluster
clus7 = cluster_map.loc[cluster_map.cluster==7].cluster
```

Figura 82 - Sep (anexo)

Fonte: elaborado pelo autor.

Transforma os tipos de dados em numérico para calcular a média de outras variáveis (u ma por vez) por cluster (Figura 82).

```
cluster_map['Global_active_power'] = pd.to_numeric(cluster_map['Global_active_power'])
cluster_map.groupby('cluster')['Global_active_power'].mean()
```

```
cluster
0    0.516187
1    1.799188
2    3.213582
3    3.980106
4    5.642811
5    4.310052
6    2.586686
7    6.183143
Name: Global_active_power, dtype: float64
```

```
[20] cluster_map.groupby('cluster')['Sub_metering_3'].mean()
```

```
cluster
0     0.406223
1    17.865676
2     0.355556
3    17.594697
4    11.270270
5    17.731988
6     0.333333
7    12.678571
Name: Sub_metering_3, dtype: float64
```

Figura 83 - Media (anexo)

Fonte: elaborado pelo autor.

Quantidade de registros por grupo (Figura 83):

```
cluster_map.groupby('cluster')['Sub_metering_3'].count()
```

```
cluster
0    12919
1     6462
2      225
3      264
4       37
5      347
6      210
7       28
Name: Sub_metering_3, dtype: int64
```

Figura 84 - Count (anexo)

Fonte: elaborado pelo autor.

Percentual de registros por cluster (Figura 84):

```
g = (cluster_map.groupby('cluster')['Sub_metering_3'].count())
s = g.sum()
c0, c1, c2, c3, c4, c5, c6, c7 = clus0.count(), clus1.count(), clus2.count(), clus3.count(), clus4.count(), clus5.c
pct = [c0/s, c1/s, c2/s, c3/s, c4/s, c5/s, c6/s, c7/s]
print('O cluster 0 tem %.2f dos registros'  %(pct[0]))
print('O cluster 1 tem %.2f dos registros'  %(pct[1]))
print('O cluster 2 tem %.2f dos registros'  %(pct[2]))
print('O cluster 3 tem %.2f dos registros'  %(pct[3]))
print('O cluster 4 tem %.3f dos registros'  %(pct[4]))
print('O cluster 5 tem %.2f dos registros'  %(pct[5]))
print('O cluster 6 tem %.2f dos registros'  %(pct[6]))
print('O cluster 7 tem %.3f dos registros'  %(pct[7]))
```

```
O cluster 0 tem 0.63 dos registros
O cluster 1 tem 0.32 dos registros
O cluster 2 tem 0.01 dos registros
O cluster 3 tem 0.01 dos registros
O cluster 4 tem 0.002 dos registros
O cluster 5 tem 0.02 dos registros
O cluster 6 tem 0.01 dos registros
O cluster 7 tem 0.001 dos registros
```

Figura 85 - Pct (anexo)

Fonte: elaborado pelo autor.

Visualização dos grupos em gráfico de dispersão, o código (Figura 86) e para comparação os dois gráficos para comparação (Figura 86 e 87):

```
clus = modelo.labels_
fig = plt.figure(figsize=(18,9))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(pca[clus == 0, 0], pca[clus == 0, 1], pca[clus == 0, 2], cmap='RdYlBu', label='Cluster 0', alpha=0.5)
ax.scatter(pca[clus == 1, 0], pca[clus == 1, 1], pca[clus == 1, 2], cmap='RdYlBu', label='Cluster 1', alpha=0.5)
ax.scatter(pca[clus == 2, 0], pca[clus == 2, 1], pca[clus == 2, 2], cmap='RdYlBu', label='Cluster 2', alpha=0.5)
ax.scatter(pca[clus == 3, 0], pca[clus == 3, 1], pca[clus == 3, 2], cmap='RdYlBu', label='Cluster 3', alpha=0.5)
ax.scatter(pca[clus == 4, 0], pca[clus == 4, 1], pca[clus == 4, 2], cmap='RdYlBu', label='Cluster 4', alpha=0.5)
ax.scatter(pca[clus == 5, 0], pca[clus == 5, 1], pca[clus == 5, 2], cmap='RdYlBu', label='Cluster 5', alpha=0.5)
ax.scatter(pca[clus == 6, 0], pca[clus == 6, 1], pca[clus == 6, 2], cmap='RdYlBu', label='Cluster 6', alpha=0.5)
ax.scatter(pca[clus == 7, 0], pca[clus == 7, 1], pca[clus == 7, 2], cmap='RdYlBu', label='Cluster 7', alpha=0.5)
ax.set_xlabel('X');ax.set_ylabel('Y');ax.set_zlabel('Z')
plt.title("Modelo do conjunto cosumo de energia elétrica")
plt.legend()
```

Figura 87 - CodDisp (anexo)

Fonte: elaborado pelo autor.

```
<matplotlib.legend.Legend at 0x7f6e0482f7f0>
```

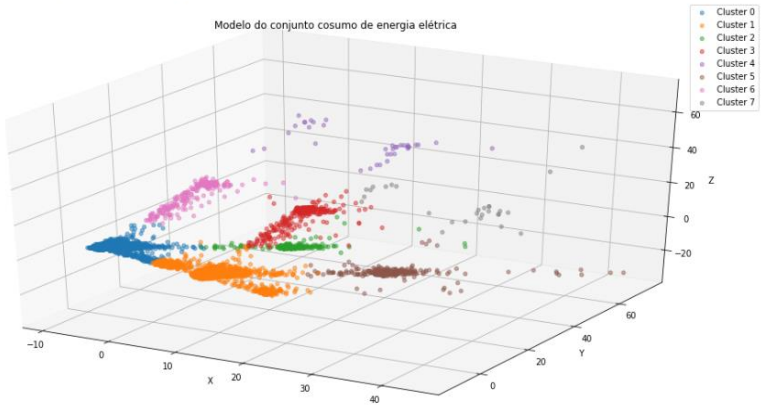


Figura 88 - GrafDisp (anexo)

Fonte: elaborado pelo autor.

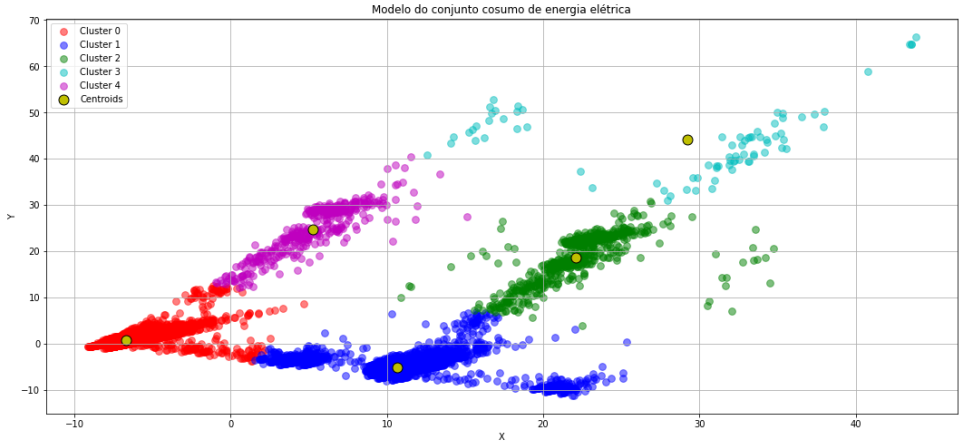


Figura 89 - GrafDisp (HPC)

Fonte: elaborado pelo autor.

10.Conclusão e trabalhos futuros

Concluo que o trabalho foi proveitoso, embora tenha tido bastante dificuldade no início do projeto com relação a pesquisar algo que já existisse em conjunto no que se refere aos dois temas que lidamos, também houve necessidade de investimento de grande tempo de estudo do algoritmo kmeans, embora seja dito nesse mesmo trabalho que é um algoritmo de fácil manipulação, nessa área de aprendizado de máquinas existe sempre a necessidade de muito estudo pois pequenos detalhes podem fazer uma grande diferença na construção do projeto. Mas apesar de tudo isso conseguimos chegar ao objetivo do projeto pois desenvolvemos a Máquina Preditiva usando o algoritmo k-means, com a função que criamos, podemos dizer que em conjunto com o número mágico (7+-2) e a partir dos resultados construímos outros *dataset's* que podem ser utilizados para aprendizagem humana; até fomos além, de forma que há mais conteúdo para aquele que tiver acesso a esse relatório e de alguma forma possa aumentar a possibilidade de aprendizagem com o mesmo, principalmente no que se refere a análise estatística para fins de auxílio em decisão.

Fica à disposição do professor Paulo Vieira, como era alvo desse trabalho, os dataset's com os respectivos números dos grupos para trabalhos futuros com aprendizagem humana, e ainda a própria máquina preditiva podendo esta ser ainda mais aperfeiçoada e assim auxiliar novos alunos nos trabalhos relacionado a aprendizado de máquinas (Machine Learning) não supervisionado.

Como trabalho futuro para mim próprio, fica o encargo de tentar aperfeiçoar esse projeto e produzir insights a partir do que se conseguiu visualizar com o trabalho realizado, para assim pudesse auxiliar numa possível tomada de decisão.

Referencias

https://colab.research.google.com

https://www.youtube.com/@RafinhadosDados

https://www.youtube.com/@CienciadosDados

https://statplace.com.br/

https://github.com

TMEMC_JoseMesdes.pdf (uab.pt)

https://www.princeton.edu/news/2012/07/26/george-miller-princeton-psychology-professor-and-cognitive-pioneer-dies

(3) ANÁLISE DE AGRUPAMENTOS NÃO SUPERVISIONADOS | LinkedIn

ulfc106238_tm_Vania_Gomes.pdf

https://archive.ics.uci.edu/ml/datasets

http://scikit-learn.org

https://www.youtube.com/watch?v=onuFe_ybTZE

https://pt.wikipedia.org

https://medium.com/pizzadedados/kmeans-e-metodo-do-cotovelo-94ded9fdf3a9

Wang, Juntao e Xiaolong Su. "Um algoritmo de agrupamento K-Means aprimorado." *2011 IEEE 3ª Conferência Internacional de Software e Redes de Comunicação* . IEEE, 2011.

Miller, G. A. (1956). "The magical number seven, plus or minus two: Some limits on our capacity for processing information". *Psychological Review*.