



UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

LUCAS SALES DE ARAÚJO NUNES

**IDENTIFICAÇÃO E RECONHECIMENTO DE PADRÕES DE
CONSUMO DE ENERGIA ELÉTRICA**

CAMPINA GRANDE, PARAÍBA

DEZEMBRO DE 2018

Lucas Sales de Araújo Nunes

IDENTIFICAÇÃO E RECONHECIMENTO DE PADRÕES DE CONSUMO DE ENERGIA ELÉTRICA

*Trabalho de Conclusão de Curso apresentado à
Coordenação do Curso de Graduação em
Engenharia Elétrica da Universidade Federal de
Campina Grande, Campus SEDE, como parte
dos requisitos necessários para a obtenção do
grau de Bacharel em Engenharia Elétrica.*

Área de Concentração: Processamento de Energia

Orientador: Edmar Candeia Gurjão

Campina Grande, Paraíba

Dezembro de 2018

Lucas Sales de Araújo Nunes

IDENTIFICAÇÃO E RECONHECIMENTO DE PADRÕES DE CONSUMO DE ENERGIA ELÉTRICA

*Trabalho de Conclusão de Curso apresentado à
Coordenação do Curso de Graduação em
Engenharia Elétrica da Universidade Federal de
Campina Grande, Campus SEDE, como parte
dos requisitos necessários para a obtenção do
grau de Bacharel em Engenharia Elétrica.*

Aprovado em: ____ / ____ / ____

Professor Avaliador

Universidade Federal de Campina Grande

Avaliador

Edmar Candeia Gurjão

Universidade Federal de Campina Grande

Orientador

Dedico este trabalho aos meus pais Francisco e Kátia e a minha irmã Ana Raquel.

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus, a força que me sustenta. Aos meus pais Francisco e Kátia, por tudo o que fizeram por mim e pela pessoa que sou hoje. Por todo o trabalho e todo o esforço sempre procurando me oferecer o melhor

À minha irmã Ana Raquel e aos meus familiares próximos, por sempre estarem ao meu lado nos momentos mais tristes e mais alegres, torcerem pelo meu sucesso e pela constante preocupação com o meu bem estar.

Agradeço aos amigos que fiz durante o curso, e pela segunda família que ganhei. Em especial à Camila, Isaque, João Carlos, Pedro e Monaliza, que foram e estão sendo meu suporte diante das dificuldades.

Agradeço aos meus orientadores Glauco Fontgalland e Edmar Gurjão, pelo aprendizado e conhecimento compartilhado e pela oportunidade de trabalhar com eles.

Por fim, agradeço aos demais professores e funcionários do DEE por contribuírem na minha formação de forma direta e indireta.

“Mera mudança não é crescimento. Crescimento é a síntese de mudança e continuidade, e onde não há continuidade não há crescimento.”

C. S. Lewis

RESUMO

Este trabalho foi desenvolvido com o objetivo de identificar e reconhecer padrões de consumo de energia elétrica. Foram aplicados modelos computacionais para agrupar e classificar esses dados de acordo com as suas características mais semelhantes. Os algoritmos utilizados foram o *k-means* (identificação) e a Máquina de Vetor de Suporte (reconhecimento), e estes atenderam o fim pretendido, apresentando bons resultados na segmentação de perfis de consumo. Como caso de uso foi utilizado os dados do consumo de energia do Bloco CJ da Universidade Federal de Campina Grande. Os algoritmos foram aplicados a dados de potência ativa correspondentes a seis semanas de medições, e mostrou-se que é possível detectar padrões de consumo utilizando os algoritmos.

Palavras-Chave: Padrões de Consumo, Modelos Computacionais, *k-means*, Máquina de Vetor de Suporte, Energia Elétrica.

ABSTRACT

This work was developed to identify and detect patterns in electrical energy consumption. Computational models were applied to group and to classify the consumption data according to similar features. The algorithms *k-means* (identification) and *Vector Support Machine* (recognition) were used in the data of energy consumption of the CJ Building of the Universidade Federal de Campina Grande. Data of active power for six weeks were analyzed, and patterns were detected.

Keywords: Consumption Patterns, Computer Models, *k-means*, Support Vector Machine, Electric Power.

LISTA DE FIGURAS

Figura 1 - Etapas do Processo de KDD.....	16
Figura 2 - Clusterização em espaço 2D. (a) Dados iniciais. (b) Formação de três clusters. (c) Formação de quatro clusters.....	19
Figura 3 - Execução do método k-means.....	22
Figura 4 - Hiperplanos de separação em SVM.....	23
Figura 5 - Máquina de Vetor de Suporte, ilustração de um classificador linear.....	24
Figura 6 - Vetores-suporte demarcando as linhas de separação entre duas classes.....	24
Figura 7 - Margem de separação entre hiperplanos nas fronteiras das classes positiva (quadrados) e negativa (círculos). Os eixos x_1 e x_2 representam as dimensões das amostras no espaço 2D.....	25
Figura 8 - Distribuição de amostras impossíveis de serem separadas por um classificador linear.....	26
Figura 9 - Exemplo de aplicação da função kernel.....	27
Figura 10 - Processo de classificação utilizando o mapeamento kernel.....	28
Figura 11 - Medidor de energia elétrica confeccionado pelo projeto SCIKE Paraíba.....	29
Figura 12 - Curvas de carga do bloco CJ entre os dias 16 e 23 de novembro.....	30
Figura 13 - Processo de desenvolvimento e execução.....	31
Figura 14 - Ambiente computacional jupyter notebook.....	32
Figura 15 - Curva de decrescimento da soma dos erros quadráticos em função do incremento de k	33
Figura 16 - Clusters obtidos para $k = 4$	34
Figura 17 - Clusters obtidos para $k = 5$	34
Figura 18 - Curvas de carga por dia da semana.....	36
Figura 19 - Matriz de Confusão.....	37
Figura 20 - Matrizes de erro para as classificações de consumo de uma segunda-feira (à esquerda) e de uma quarta-feira (à direita).....	38

LISTA DE TABELAS

Tabela 1 - Tarefas para descoberta de conhecimento e suas aplicações.	18
Tabela 2 - Padrões de carga por dia da semana.....	35
Tabela 3 - Matriz de Erro de Base de Treinamento.	37

LISTA DE ABREVIATURAS

AG	<i>Acurácia Global</i>
DR	<i>Disjuntor Diferencial Residual</i>
KDD	<i>Knowledge-Discovery in Databases</i>
SVM	<i>Support Vector Machine</i>
W	<i>Watts</i>
IoT	<i>Internet of Things</i>

Sumário

1	INTRODUÇÃO	13
1.1	Objetivos	14
1.1.1	Objetivos Específicos	14
1.2	Estrutura do Trabalho.....	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Medidores Inteligentes	15
2.2	Descoberta de Conhecimento em Base de Dados	15
2.2.1	Algoritmos de Agrupamento	19
2.2.1.1	Formulação Matemática	20
2.2.2	Máquina de Vetores de Suporte	22
2.2.2.1	Funções <i>Kernel</i>	26
3	MATERIAIS E MÉTODOS	29
3.1	Projeto SCIKE Paraíba.....	29
3.2	Aquisição e Visualização de Dados	29
3.3	Modelos Computacionais	30
3.4	Processamento de Dados.....	30
3.5	Metodologia de Desenvolvimento e Execução	31
4	RESULTADOS E DISCUSSÕES	33
4.1	Análise de <i>Clustering</i> e Identificação de Padrões	33
4.2	Reconhecimento de Padrões.....	36
4.2.1	Avaliação do Classificador.....	36
5	CONCLUSÕES.....	39
	REFERÊNCIAS BIBLIOGRÁFICAS	40
	APÊNDICE A	42
	APÊNDICE B	46

1 INTRODUÇÃO

A crescente oferta e o interesse em automação têm possibilitado uma maior comodidade para os consumidores e conquistado o mercado. O desenvolvimento de dispositivos inteligentes e as facilidades de comunicação permitem a obtenção de dados de maneira rápida e em tempo real. Tais equipamentos geram uma grande quantidade de dados, e o seu processamento para obter informação requer soluções eficientes e automatizadas.

O monitoramento não invasivo de cargas elétricas é uma área de constante pesquisa e de grande interesse pela comunidade acadêmica (DANTAS, 2016). Com um medidor de energia instalado no painel de entrada de um sistema, pode-se estimar o consumo das cargas que o compõem. No âmbito das redes inteligentes (*smart grids*), é possível reduzir significativamente os custos das medições e permitir mais possibilidade de gerenciamento. Já o conceito de Internet das Coisas (*Internet of Things-IoT*), aumenta a viabilidade de conexão entre os dispositivos.

Mesmo com o aumento do fluxo de dados em diversos setores, consumidores ainda participam de forma passiva em relação ao uso de energia elétrica. Os medidores inteligentes atuais (*smart meters*) mostram apenas o consumo total de energia, sem apresentar detalhes de como esta é consumida. A discriminação do uso de cargas elétricas em unidades consumidoras de forma individual, permite ao consumidor conhecer seu padrão de consumo e usar a energia de forma mais eficiente (MONZANI, 2016).

O desenvolvimento de métodos de agrupamento baseados no reconhecimento de padrões de consumo é importante no gerenciamento da eficiência do setor elétrico. De forma geral, os métodos de agrupamento, ou de reconhecimento de padrões aplicados ao consumo de energia elétrica, se propõem a identificar os padrões de consumo intrínsecos aos grupos de uma amostra de dados a partir da extração de características desta última (FERREIRA, 2016), resultando na obtenção de grupos semelhantes entre si (homogêneos) e com altas dissimilaridades (heterogeneidade) entre eles, permitindo assim um melhor gerenciamento do consumo de energia.

A diferença entre as demandas máximas e mínimas, consumidas ao longo de um dia, apresenta grande variação de acordo com o comportamento típico de determinada unidade consumidora. Portanto, faz-se necessário ter o conhecimento das curvas de carga diárias típicas dos consumidores.

1.1 Objetivos

O presente trabalho tem como objetivo a identificação e o reconhecimento dos perfis dinâmicos de consumo de energia.

1.1.1 Objetivos Específicos

a) Realizar análises de *clustering* para identificar os padrões de consumo semelhantes ao longo do tempo e agrupá-los em grupos correspondentes,

b) Utilizar algoritmos classificadores para o reconhecimento desses padrões.

c) Obter curvas típicas de carga por dia da semana (dias úteis) e a classificação do consumo em patamares (alto, médio e baixo).

d) Ajudar na detecção de comportamentos anômalos, o que é de grande importância na tomada de decisões que visam a economia e redução de custos.

1.2 Estrutura do Trabalho

O Capítulo 1 apresenta, de forma introdutória, a contextualização do tema abordado, definição dos objetivos e apresentação da estrutura do trabalho.

Em seguida, no Capítulo 2, tem-se a fundamentação teórica, apresentando conceitos como o de medidores inteligentes e descoberta de conhecimento em bases de dados.

O Capítulo 3 traz os materiais e a metodologia utilizada para o desenvolvimento do trabalho. Em seguida os resultados e as discussões são apresentados no Capítulo 4. O Capítulo de conclusão destaca os principais entendimentos do trabalho e sugestões de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Medidores Inteligentes

O conceito de medição inteligente vem se consolidando no sentido de melhorar gestão e o uso eficiente da energia elétrica (SOUZA et al., 2017). Nesse contexto, há um grande interesse em desenvolver tecnologias que possibilitem aos consumidores o uso racional desse recurso. Diante disso, novos medidores de energia elétrica vêm sendo desenvolvidos para prover informações detalhadas sobre o consumo. Tais dispositivos são conhecidos como medidores inteligentes de energia (*smart meters*) e possibilitam maior exatidão de medição, capacidade de armazenamento, envio automático de informações de consumo e integração com sistemas de gerenciamento de energia (SOUZA et al, 2017).

Dos *smart meters* destacam-se as seguintes características (ZHENG et al, 2013):

- Qualidade e clareza das informações coletadas;
- Disponibilização de informações que impactem diretamente sob o comportamento do consumidor;
- Possibilidade de consulta dos consumos anteriores, ou ainda das médias de consumo;
- Previsões baseadas no histórico comportamental do perfil do consumidor;
- Controle de gastos.

Atualmente, um dos principais temas destacados na literatura é a utilização dos medidores inteligentes de energia para realizar a desagregação do consumo por cargas. Tal função também é destacada como um aprimoramento dos *smart meters*, denominado medidor cognitivo de energia (SOUZA et al., 2017).

Entre suas principais aplicações, destacam-se: o monitoramento do consumo de energia em ambientes residenciais e a criação de mecanismos que possibilitem aos consumidores entenderem seus hábitos de consumo, para posterior modificação dos mesmos e, com isso, reduzirem gastos com energia elétrica.

2.2 Descoberta de Conhecimento em Base de Dados

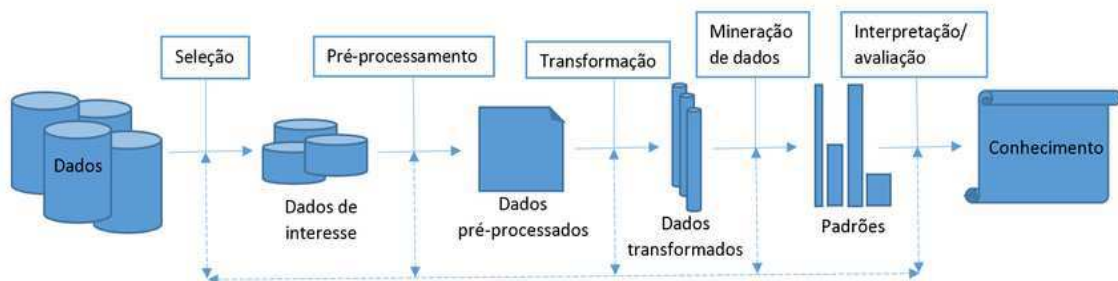
Com o desenvolvimento dos sistemas computacionais e o aumento da capacidade de armazenamento de informação em banco de dados surge a necessidade de novas técnicas e ferramentas de análise de grandes volumes de dados. As ferramentas para a descoberta em

conhecimento de base de dados integram a tecnologia de aprendizagem organizacional em busca de uma gestão estratégica de conhecimento.

Na década de 80 foi formalizado o termo KDD (*knowledge-discovery in databases*), em referência ao amplo conceito de procurar conhecimento a partir de bases de dados (FAYYAD et al, 1996). A técnica é composta de várias etapas, entre as quais destaca-se a etapa de Mineração de Dados, a qual envolve o processo de construção de modelos, que podem ser baseados em algoritmos, para determinação de padrões, a partir de dados observados e informações disponíveis.

Os algoritmos para Mineração de Dados realizam o papel de inferir conhecimento. No entanto, o processo de KDD não se resume somente na etapa de mineração, mas também abrange a preparação dos dados (aquisição, limpeza, pré-processamento, seleção, redução e projeção) e a fase de interpretação e a avaliação do modelo descoberto e um possível retorno a algum passo anterior como também uma possível visualização do modelo extraído (MARIN, 2014). As etapas do processo de KDD são representadas na Figura 1.

Figura 1 - Etapas do Processo de KDD



Fonte: (FREITAS et al, 2015).

Como observado na Figura 1, o processo de KDD é composto de várias etapas, em que diversas técnicas podem ser aplicadas. O primeiro passo é o entendimento do domínio da aplicação, definindo os conhecimentos prévios dos dados e o conhecimento que se deseja adquirir. Na extração retira-se um conjunto de dados e seleciona-se o conjunto de variáveis necessárias. Então faz-se a limpeza e o pré-processamento (remoção de dados inválidos, ruídos, dados faltantes, etc.) e a transformação dos dados preparando-os para a próxima etapa. Após a escolha da função de Mineração de Dados (classificação, sumarização, clusterização, regressão, regras de associações) e a escolha do algoritmo que será utilizado na descoberta de padrões, a geração de padrões é realizada. E por fim, na avaliação dos resultados são

interpretados os padrões encontrados a fim de identificar conhecimentos reais e interessantes para a aplicação desejada.

A Mineração de Dados é uma das etapas do processo de KDD, conforme pode ser observado na Figura 1. Dentro dessa etapa podem ser aplicadas várias técnicas e cada uma destas e algoritmos executam determinadas tarefas, conforme estão descritas, a seguir, na Tabela 1.

Tabela 1 - Tarefas para descoberta de conhecimento e suas aplicações.

Tarefa de KDD	Exemplos de Aplicações
Classificação	Permite associar a cada registro de um banco de dados um único rótulo ou classe. Uma comercializadora de energia elétrica pode realizar planos de atendimento diferenciados, conforme a segmentação dos perfis de crédito ou de montante contratual de seus clientes.
Clusterização	Possibilita a identificação automática dos registros de uma base de dados na forma de agrupamento com características similares. Neste caso, ao contrário da tarefa de classificação, não se tem o conhecimento prévio das classes existentes.
Detecção de Desvios	Detecção de fraudes em cartões de crédito, planos de saúde, arrecadação. Identificação de furtos de energia elétrica.
Previsão de Séries Temporais	Solução de problemas relacionados ao planejamento e a tomada de decisões, reduzindo os níveis de incerteza e os riscos do negócio.
Regressão	Busca de funções, lineares ou não, que mapeiem os registros de um bando de dados. Pode-se pré-estabelecer a variação da carga num determinado período em função da variação da temperatura.
Sumarização	Identificar as características dos consumidores de energia elétrica de determinado segmento industrial em relação à elasticidade-preço.

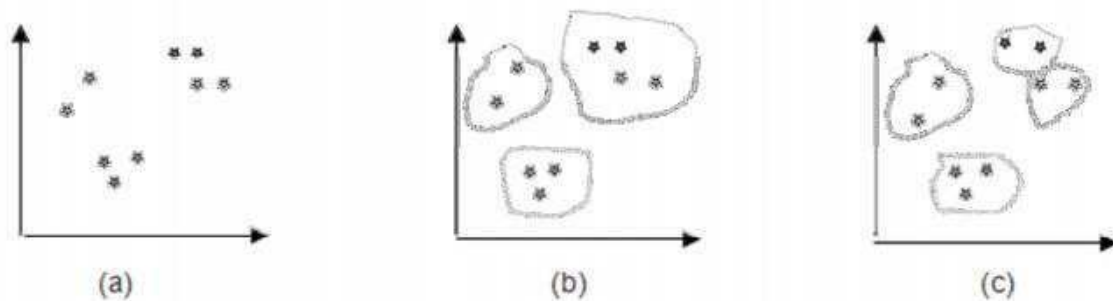
Fonte: Adaptado de (FAYYAD et al., 1996).

2.2.1 Algoritmos de Agrupamento

Agrupar é uma tarefa de KDD, normalmente confundida com classificação. Enquanto esta seleciona uma entidade baseado em valores pré-definidos de atributos, a clusterização agrupa dados similares não baseados em valores pré-definidos, mas de acordo com suas semelhanças. O método de agrupamento demonstra ser particularmente mais apropriado para a exploração de interrelações entre amostras com o intuito de realizar uma avaliação prévia da estrutura das amostras (KANTARDZIC, 2011).

A Figura 2 apresenta um exemplo de clusterização de pontos em um espaço euclidiano de duas dimensões. Considerando que o número de grupos não é previamente conhecido, a Figura 2b e 2c apresentam duas possíveis formações de clusters, a partir dos dados apresentados na Figura 2a. Nota-se que os clusters da Figura 2 podem ser reconhecidos visualmente, entretanto, isso se torna impossível em um espaço euclidiano de n-dimensões, havendo necessidade do uso de algoritmos especiais para esta tarefa.

Figura 2 - Clusterização em espaço 2D. (a) Dados iniciais. (b) Formação de três clusters. (c) Formação de quatro clusters.



Fonte: (KANTARDZIC, 2011).

Dos algoritmos de clusterização presentes na literatura, aplicados ao problema de detecção de padrões de consumo de energia elétrica, destaca-se o algoritmo *k-means* (COSTA et al., 2011), pela simplicidade tanto na programação quanto na execução de comandos. É prático, não requerendo muita memória ou tempo de processamento e sua formulação matemática é concisa.

Os avanços dos recursos computacionais fomentaram a aplicação das técnicas de algoritmos de agrupamentos em diversas áreas do conhecimento, sobretudo, na Mineração de Dados que consiste em obter conhecimentos que podem apoiar em processos decisórios.

A utilização de análise de *clustering* para agrupamentos de consumidores por perfis de carga de energia elétrica, foi demonstrada por (PRAHASTONO et al., 2008) devido a uma nova legislação, a qual permite que diferentes regiões tenham possibilidade de aplicar suas próprias regras tarifárias. Até então essas regras eram desenvolvidas por meio do uso de tabelas com dados coletados ao longo dos anos. Estes dados agrupavam os consumidores por variações em seus perfis de carga, como industriais, comerciais e residenciais.

A determinação dos perfis de carga tem sido motivada pelos mais diversos fatores, como por exemplo, o enquadramento de consumidores em novas categorias de consumo, apresentando uma nova abordagem sobre o tema, validando a análise de clusterização pelo método *k-means* (ALBERT, 2013).

Albert (2013) propõem uma análise descritiva a partir de dados desagregados de consumo, com o objetivo de oferecer à indústria energética a capacidade de agrupar usuários de acordo com semelhanças em seus padrões de consumo e utilizar características inferidas destes, para identificar comportamentos e estilo de vida dos moradores.

E por fim, Ramos e Vale (2008) apresentam uma abordagem para o agrupamento de consumidores de média tensão, com o intuito de identificar aspectos que causam aumentos dos picos de consumo, em que foi capaz de produzir perfis de carga diferenciados.

2.2.1.1 Formulação Matemática

Os algoritmos de agrupamento de objetos efetuam o reconhecimento dos grupos e dos padrões utilizando duas abordagens de agrupamentos: a hierárquica e a não hierárquica. Vale reiterar que a abordagem de agrupamento hierárquica realiza a partição dos objetos de forma gradativa e sequencial, enquanto que na abordagem não-hierárquica, há uma predefinição do número de grupos a serem obtidos, e que em geral, o método hierárquico é usado para estimar o número de grupos ótimos (validado por meio de um índice de qualidade) e, depois com esta informação, aplica-se o método não-hierárquico para obter os grupos finais. O nível de coesão/homogeneidade dentro dos grupos e de heterogeneidade entre eles é quantificado através de índices específicos de qualidade do agrupamento.

A abordagem não hierárquica é um processo dinâmico e interativo de formação de grupos. Uma vez especificado o número de grupos, o algoritmo tem como objetivo identificar ou reconhecer a melhor distribuição dos objetos conforme a premissas de homogeneidade intra e heterogeneidade intergrupos. Os procedimentos não hierárquicos compreendem

métodos que têm como objetivo a obtenção de uma partição de n elementos em c ($c \geq 2$) grupos gerando também, como resultado, uma matriz de partição $U(c \times n)$ tal que u_{ik} refere-se à pertinência do objeto k ($k = 1, \dots, n$) ao grupo i ($j = 1, \dots, c$). Cada coluna da matriz de partição fornece os graus de aderência ou pertinência de um dado objeto a todos os grupos reconhecidos.

Dentre os métodos de partição não hierárquica, os métodos baseados nos modelos *c-means* (tendência central da média aritmética) são os mais utilizados e validados quanto à eficiência e aplicabilidade em problemas de agrupamento de protótipo pontual (BEZDEK et al., 2005).

Os métodos baseados em modelos *c-means* compreendem o seguinte problema de otimização:

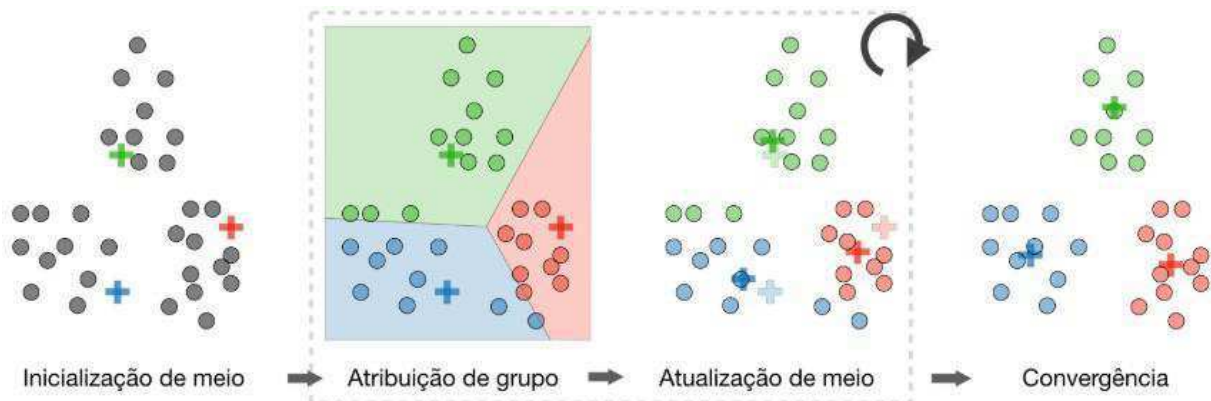
dada uma amostra de n objetos $X = \{x_1, \dots, x_n\}$, de dimensionalidade p e c grupos:

$$\text{minimizar} \left\{ J(c, u) = \sum_{i=1}^m \|x_i - u_{c_i}\|^2 \right\} \quad (1)$$

Sendo $J(c, u)$ a função de distorção, que corresponde à soma das distâncias mínimas entre as amostras de dados e os centróides selecionados.

O método *k-means*, de acordo com (BEZDEK et al., 2005), forma k grupos a partir de uma amostra em que cada objeto pertença a apenas um dos grupos. As etapas do procedimento do algoritmo são (Figura 3): determinação aleatória de k objetos para serem os primeiros centróides dos agrupamentos; associação de cada um dos $(n - k)$ objetos ao centróide mais próximo através da distância euclidiana para cada centróide; e recálculo de um novo centro com base nos elementos associados a cada centróide. Estes passos são repetidos até que ocorra a convergência do agrupamento. Esse algoritmo sofre influência da diferença de escala entre os valores dos atributos, e sendo assim, a normalização das variáveis é uma ação preliminar recomendável.

Figura 3 - Execução do método k-means.



Fonte: <https://stanford.edu/~shervine/l/pt/teaching>.

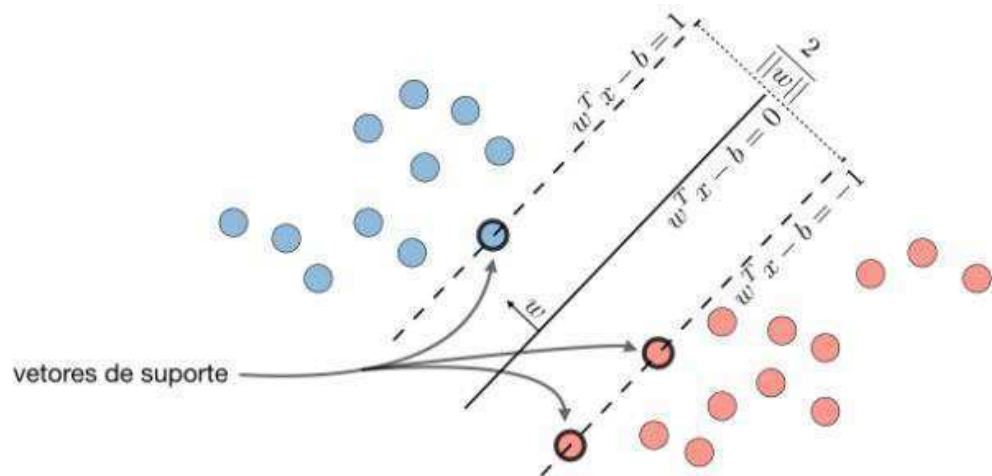
O algoritmo *k-means* funciona com a premissa de que o usuário seleciona o número de clusters desejados a fim de se caracterizar determinado conjunto de dados. Nesse caso, é necessária a utilização de uma ferramenta que valide e interprete os resultados de modo adequado. O “*Elbow Method*” ou “Método do Cotovelo”, denominado assim por sua curva assemelhar-se a um formato de braço flexionado, é utilizado para este fim. De posse dos dados que se deseja padronizar, o método se baseia na soma dos erros quadráticos para um número ideal de clusters: varia-se o valor de k (número de clusters) a partir de 1 e analisa-se a melhoria do resultado a cada incremento. Quando o benefício deixa de ser relevante (um salto entre uma quantidade de cluster e a próxima), a diferença da distância é quase insignificante. A partir desse momento entende-se que o algoritmo é relevante com aquela quantidade k e então ele pode ser utilizado para segmentar os dados.

2.2.2 Máquina de Vetores de Suporte

Fundamentada na Teoria do Aprendizado Estatístico, a técnica denominada Máquina de Vetores de Suporte foi proposta com o objetivo de resolver problemas na área de reconhecimento de padrões e regressão estatística. A ideia básica da SVM (*Support Vector Machine*) consiste em identificar um hiperplano que possa ser empregado como superfície de decisão, de tal forma que a margem de separação entre as classes de dados seja máxima (GASPAR, 2015), em que pode-se considerar que uma separação adequada é conseguida através do hiperplano mais distante em relação a todas as instâncias do conjunto de testes.

A Figura 4 apresenta, à esquerda, um hiperplano de separação com uma margem pequena e, à direita, um hiperplano de separação com margem máxima para as classes de dados consideradas.

Figura 4 - Hiperplanos de separação em SVM.



Fonte: <https://stanford.edu/~shervine/l/pt/teaching>

A implementação de um algoritmo de SVM baseia-se no produto interno entre um vetor de suporte x_i e um vetor x extraído do espaço de entrada. Os vetores de suporte representam um pequeno subconjunto de dados de treinamento selecionado pelo algoritmo. Dependendo de como esse produto interno é gerado, é possível definir diferentes máquinas de aprendizagem, caracterizadas por superfícies de decisões não lineares e próprias (KINTO, 2011).

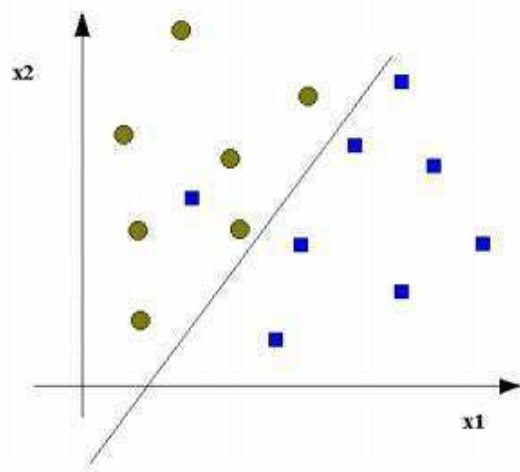
A SVM é um classificador linear cuja função f discriminante geral é da forma:

$$f(x, w, b) = \text{sign}(w \cdot x + b) \quad (2)$$

Sendo, $\text{sign}(q) = 1$, se $q \geq 0$ e $\text{sign}(q) = -1$, se $q < 0$; b é o *bias*, parâmetro que permite que o hiperplano separador se posicione no local correto para a separação das amostras.

Usando-se a equação (2), separam-se dois espaços, cada um deles identificado como 1 (círculos na parte superior) ou -1 (quadrados na parte inferior), como pode ser visto na Figura 5, ilustrando um SVM linear, separando as classes de elementos com um certo grau de ruído.

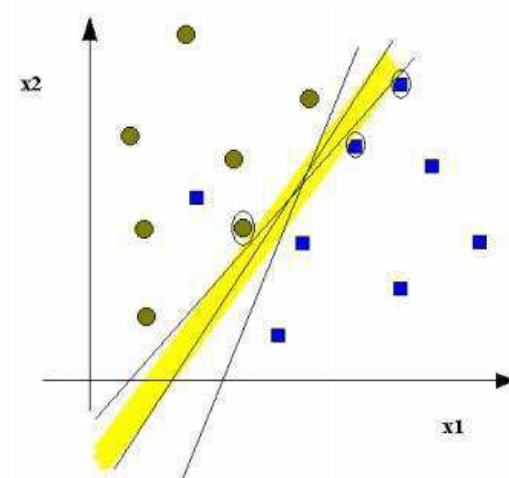
Figura 5 - Máquina de Vetor de Suporte, ilustração de um classificador linear.



Fonte: (KINTO, 2011).

Existem diversas retas (Figura 6) que poderiam estar no lugar da apresentada na Figura 4. Sendo assim, a SVM visa encontrar a maior margem de separação entre as classes pretendidas. Os vetores-suporte são as amostras limítrofes da margem e servem como fronteira na sua maximização. Deseja-se então, que o algoritmo tenha uma boa capacidade de generalização, escolhendo-se então a melhor função discriminadora dentre todas as possíveis.

Figura 6 - Vetores-suporte demarcando as linhas de separação entre duas classes.

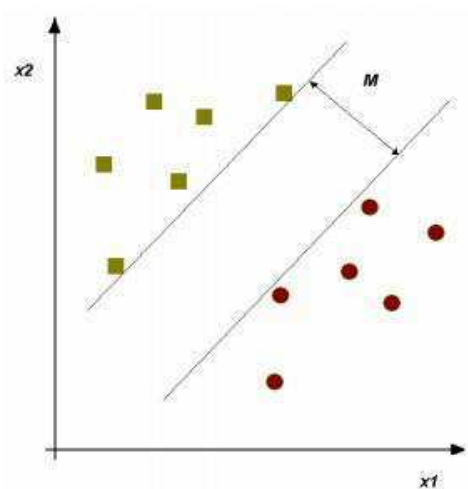


Fonte: (KINTO, 2011)

Observamos na Figura 5 que existem várias possibilidades de separar linearmente duas classes, visto isso, o objetivo da SVM é trabalhar com margem de separação ao invés de trabalhar apenas com superfície ou hiperplano (KINTO, 2011).

A maximização da margem M de separação entre as superfícies positivas e negativas pode ser vista na Figura 7.

Figura 7 - Margem de separação entre hiperplanos nas fronteiras das classes positiva (quadrados) e negativa (círculos). Os eixos x_1 e x_2 representam as dimensões das amostras no espaço 2D.



Fonte: (KINTO, 2011)

Dado que, no espaço 2D:

- O semiplano positivo é separado pela reta, $\{x_p: w^T \cdot x_p + b = 1\}$
- O semiplano negativo é separado pela reta, $\{x_n: w^T \cdot x_n + b = -1\}$
- $x_p = x_n + \alpha w$

Os parâmetros w e b são, respectivamente, o peso e o *bias*, que serão ajustados durante o processo de treinamento da SVM, x_p representa as amostras de classe positiva e x_n as de classe negativa.

Pode-se calcular a margem M em função de w e b , como a seguir:

$$M = |x_p - x_n|$$

$$M = |\alpha w|$$

$$M = \alpha \sqrt{w^T \cdot w} \quad (3)$$

Mas,

$$w \cdot (xn + \alpha w) + b = 1$$

$$w \cdot xn + b + \alpha w^T \cdot w = 1$$

$$\alpha w^T \cdot w = 2$$

$$\alpha = \frac{2}{\sqrt{w^T \cdot w}} \quad (4)$$

Resolver um problema usando SVM resume-se, então, a minimizar uma função objetiva

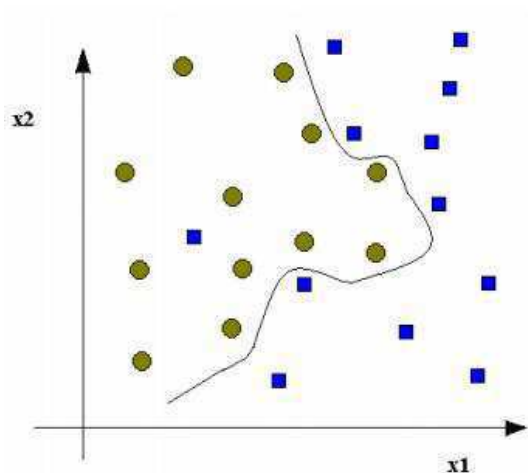
$$f: f \frac{1}{2} \|w\|^2 \quad (5)$$

Sendo f uma função quadrática.

2.2.2.1 Funções *Kernel*

Existem situações em que as amostras a serem separadas pela SVM não estão bem “comportadas” como ilustra a Figura 8. Nesse caso é necessário usar um classificador com função de separação não linear. O quadrado entre os círculos representa uma amostra com características diferentes do grupo ao qual elas pertencem; essas amostras são tratadas, dentro do possível, com parâmetros de ajuste no treinamento do algoritmo.

Figura 8 - Distribuição de amostras impossíveis de serem separadas por um classificador linear.



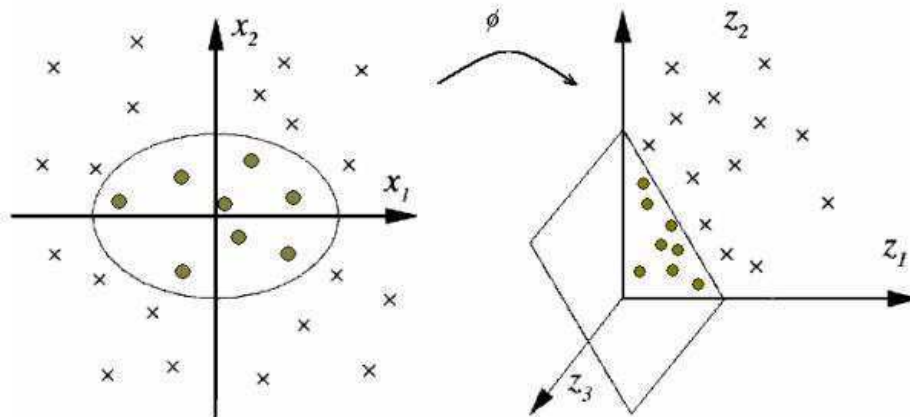
Fonte: (KINTO, 2011).

Uma SVM linear poderá não ser capaz de separar amostras de um domínio usando um semiplano. Nessas situações pode-se trabalhar um novo domínio para o qual as amostras de entrada são mapeadas. Esse novo domínio chamado de espaço das características permite o uso de um classificador linear.

A Equação 5 está usando uma função *kernel* linear, que nada mais é que um produto escalar entre dois vetores. Por intermédio dela é possível trabalhar no espaço das transformadas onde se consegue realizar uma separação linear entre as amostras das duas classes.

A Figura 9 ilustra o conceito do mapeamento das amostras do espaço das entradas para o espaço das transformadas. À esquerda apresentam-se amostras não separáveis por uma reta. Já as mesmas são separáveis no espaço das transformadas por um hiperplano (à direita).

Figura 9 - Exemplo de aplicação da função kernel.



Fonte: (SCHOLKOPF et al., 2002).

A função *kernel* linear (Equação 7) é usada na SVM linear onde as classes são linearmente separáveis. Já as funções polinomial e RBF, respectivamente Equações 8 e 9, são utilizadas quando não se consegue uma separação linear das amostras.

$$\text{Kernel Geral: } K(x, x') = \phi(x)^T \phi(x') \quad (6)$$

$$\text{Kernel Linear: } K(x, x') = (x^T \cdot x) \quad (7)$$

$$\text{Kernel Polinomial: } K(x, x') = (1 + (x^T \cdot x))^p \quad (8)$$

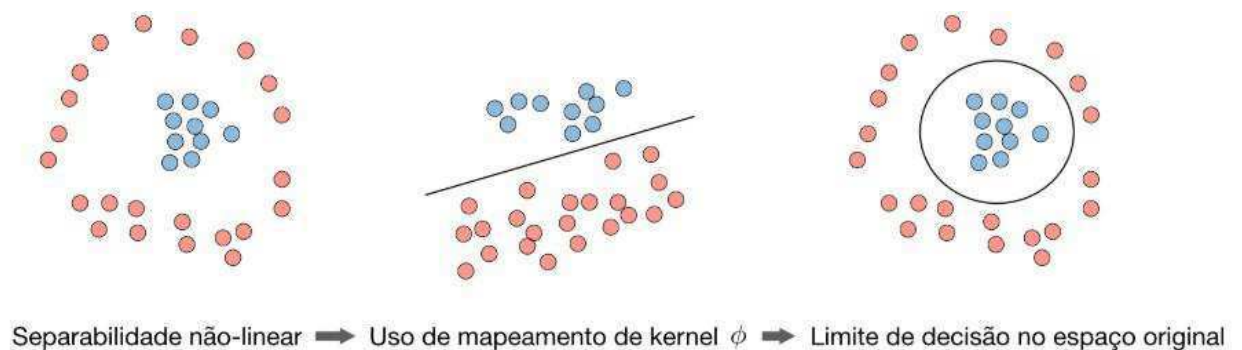
Sendo p o grau da função polinomial.

Kernel de base gaussiana radial, *radial basis function* – RBF:

$$K(x, x') = \exp\left(-\frac{1}{2\gamma} \|x - x'\|^2\right) \quad (9)$$

Sendo γ a largura da função de base radial gaussiana e x' a variável mapeada. A Figura 10 ilustra o processo de classificação utilizando funções *kernel*.

Figura 10 - Processo de classificação utilizando o mapeamento *kernel*.



Fonte: <https://stanford.edu/~shervine/l/pt/teaching>

Algumas das principais características das SVMs que tornam seu uso atrativo são (SCHOLKOPF et al., 2002):

- Possui uma base teórica já bem estabelecida nos campos da Matemática e Estatística;
- Os classificadores gerados por SVM tendem a alcançar bons resultados em termos de generalização. Essa capacidade é medida por sua eficiência na classificação de dados que não pertencem ao conjunto utilizado em seu treinamento. Na geração de preditores por SVMs, assim como em outras abordagens para o reconhecimento de padrões, é evitado o *overfitting*, situação na qual o preditor se torna muito especializado no conjunto de treinamento, obtendo baixo desempenho quando confrontado com novos padrões;
- Abordagens por SVM, em geral, apresentam elevada robustez quando utilizadas em problemas de grandes dimensões, como por exemplo, no tratamento de imagens.

3 MATERIAIS E MÉTODOS

3.1 Projeto SCIKE Paraíba

A ideia de realização deste trabalho é originada do projeto SCIKE Paraíba, uma parceria firmada entre empresas alemãs do convênio SCIKE, a Universidade Federal da Paraíba (UFPB) e a Universidade Federal de Campina Grande (UFCG). Esse projeto propõem um sistema para gerenciar a utilização de energia da UFPB e UFCG, através da coleta de informações do consumo de energia de cada bloco, por meio de sensores que medem parâmetros como potência ativa reativa e fator de potência, que são transmitidos para um servidor de armazenamento de dados (SCIKE, 2018). De posse destes, serão realizadas análises de forma a propor melhorias na gestão de consumo de energia elétrica.

3.2 Aquisição e Visualização de Dados

Para a obtenção dos dados de consumo de energia elétrica referentes ao bloco CJ, usados como treinamento e teste dos algoritmos de identificação e reconhecimento de padrões, foi utilizado um medidor inteligente, como ilustrado na Figura 11, composto por 4 partes: placa de aquisição, um microcontrolador (Arduino Uno), um chip ESP8266 e um Disjuntor Diferencial Residual (DR).

Figura 11 - Medidor de energia elétrica confeccionado pelo projeto SCIKE Paraíba.

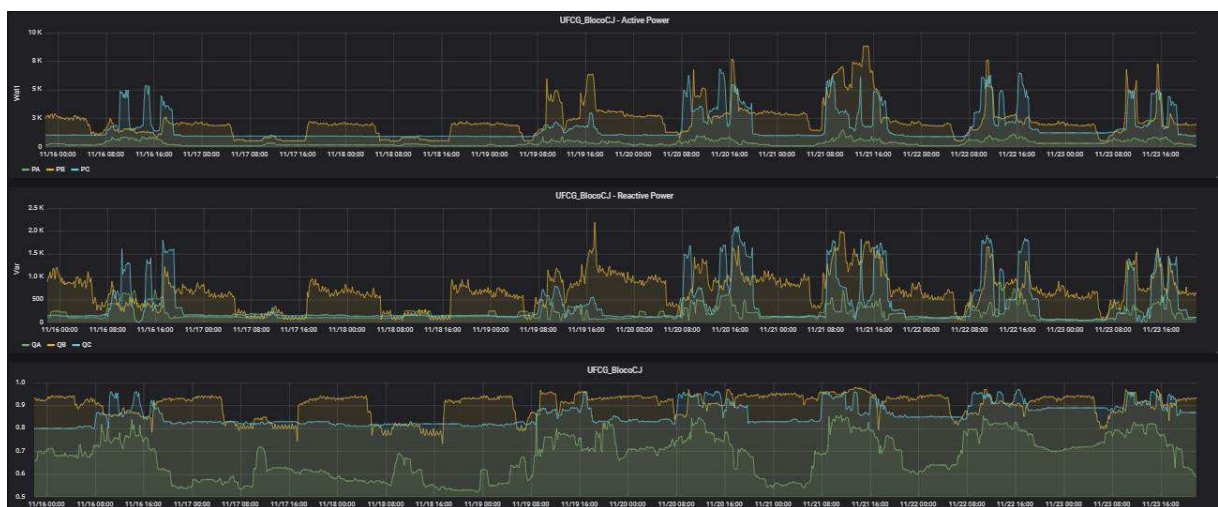


O medidor é trifásico e utiliza uma placa de montagem capaz de medir a tensão e seu ângulo instantâneo, a corrente e seu ângulo instantâneo e o fator de potência de cada uma das três fases. O chip que calcula esses parâmetros é o Atmel M90E36A. O microcontrolador é responsável pela leitura das saídas do chip, realizando correções quando necessário, e calcula

os valores médios da potência ativa, reativa e o fator de potência de cada fase, por minuto, e os envia para o ESP8266. Este componente tem a função de encaminhar os dados recebidos pelo Arduino, para um servidor local de armazenamento de dados através da rede Wi-Fi.

A visualização e análise dos dados em tempo real é viabilizada pela utilização do Grafana, uma suíte de código aberto que permite este fim. A Figura 12 apresenta os dados de consumo de potência ativa e reativa, e fator de potência do bloco CJ dos dias 16 a 23 de novembro de 2018.

Figura 12 - Curvas de carga do bloco CJ entre os dias 16 e 23 de novembro.



3.3 Modelos Computacionais

Para a elaboração dos modelos computacionais foi utilizada a plataforma de distribuição Anaconda, das linguagens de programação Python e R, que visa simplificar o gerenciamento e a execução de pacotes. Gratuita e de código aberto, amplamente usada em aplicativos de ciência de dados e aprendizagem de máquina.

Desse modo, foram implementados os algoritmos *k-means* (para a segmentação dos dados em grupos de características semelhantes) e Máquina de Vetor de Suporte (para o reconhecimento desses padrões), em Python, com o auxílio da biblioteca *scikit-learn*. Esta inclui vários outros algoritmos de classificação, regressão e agrupamento, que interagem com outras bibliotecas numéricas e científicas como *Pandas*, *NumPy*, *Seaborn* e *Matplotlib*.

3.4 Processamento de Dados

O Grafana permite a importação dos arquivos de medição em várias extensões, dentre elas a “.csv”, na qual foram extraídos os dados de interesse: consumo de potência ativa por

fase nos dias úteis da penúltima semana de outubro, de todo o mês de novembro e da primeira semana de dezembro.

O tratamento de dados (organização, padronização e remoção de *outliers* e de medições nulas) foi realizado com o auxílio do software Microsoft Office Excel 2016 ® e das bibliotecas numéricas da linguagem Python *Pandas*, *NumPy*, *Seaborn* e *Matplotlib*.

3.5 Metodologia de Desenvolvimento e Execução

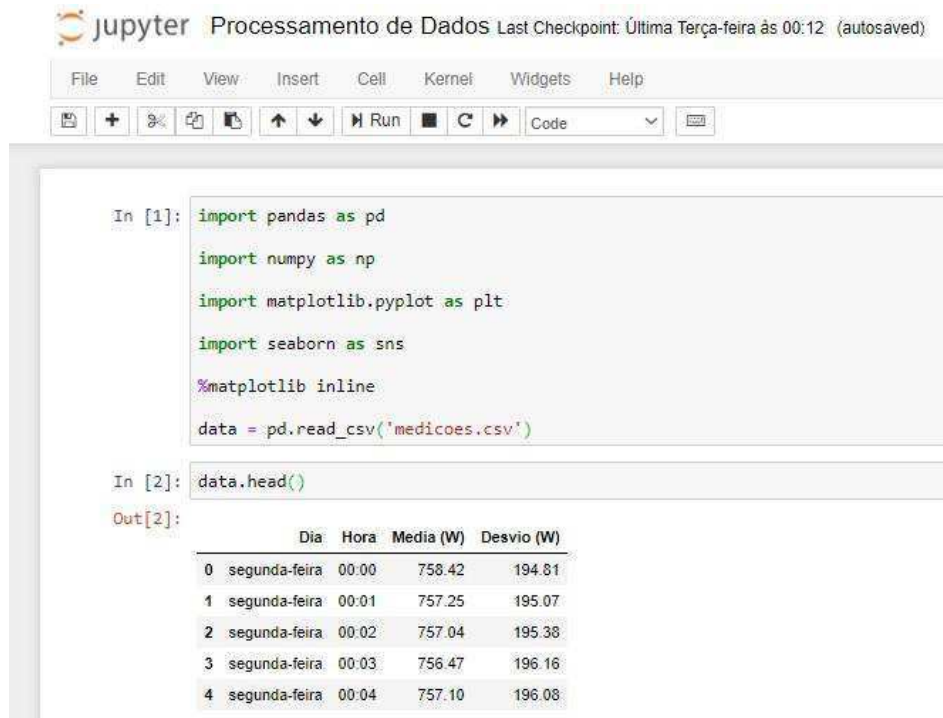
Os dados importados da base de dados do Scike para o Excel possuem as informações de potência ativa consumida (em cada fase) por minuto, bem como a data e a hora de cada medição. Logo após, foram dispostos em colunas por dia útil da semana (dados referentes às medições realizadas das segundas-feiras até as sextas-feiras). Em seguida, calculou-se a média e o desvio padrão do consumo diário de seis semanas (de uma fase), em W (watts), obtendo-se cinco tabelas com informações sobre dia, hora, média e desvio, servindo assim como dados de entrada para o processamento em Python. O processo é ilustrado na Figura 13.

Figura 13 - Processo de desenvolvimento e execução.



Utilizando o *Jupyter Notebook*, um ambiente computacional interativo da plataforma Anaconda (Figura 14), e as bibliotecas numéricas e científicas da linguagem Python, foram importados os arquivos criados anteriormente e estes foram convertidos em *data frames* (matrizes com colunas nomeadas) para viabilizar a visualização e a manipulação dos dados no jupyter.

Figura 14 - Ambiente computacional jupyter notebook.



The screenshot shows a Jupyter Notebook interface with the title 'Processamento de Dados' and a last checkpoint of 'Última Terça-feira às 00:12 (autosaved)'. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, cell navigation, and execution. The notebook content consists of two input cells and one output cell. The first input cell contains Python code for importing libraries and reading a CSV file. The second input cell contains a command to view the first five rows of the data. The output cell displays a table with five rows of data.

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
data = pd.read_csv('medicoes.csv')
```

```
In [2]: data.head()
```

```
Out[2]:
```

	Dia	Hora	Media (W)	Desvio (W)
0	segunda-feira	00:00	758.42	194.81
1	segunda-feira	00:01	757.25	195.07
2	segunda-feira	00:02	757.04	195.38
3	segunda-feira	00:03	756.47	196.16
4	segunda-feira	00:04	757.10	196.08

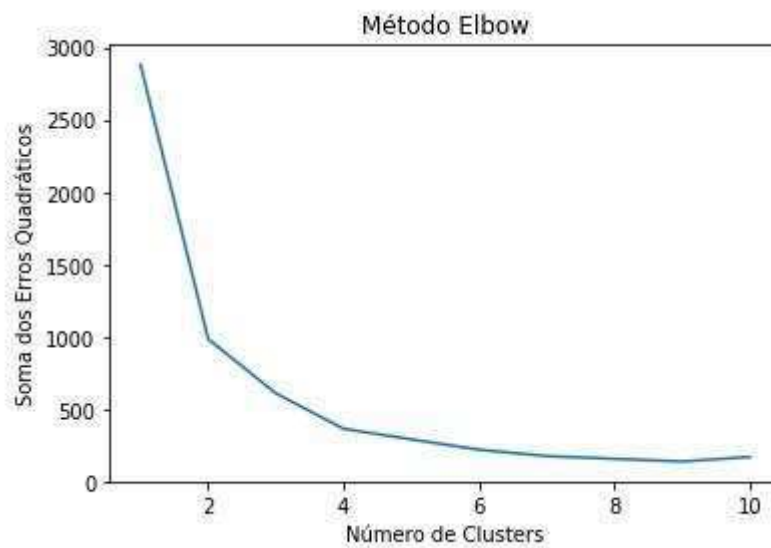
Muitos algoritmos de *Machine Learning* assumem que os dados estão padronizados e que não possuem valores nulos no momento em que geram o modelo. Quando isso não acontece, alguns tendem a possuir um desempenho ruim, principalmente em algoritmos que se baseiam no cálculo de distâncias. Por isso, foram utilizadas as funções *StandardScaler()* e *Imputer()* da biblioteca *scikit-learn*, que ignora a forma original da distribuição e transforma os dados para valores com médias em próximas de zero e com desvio padrão em torno de um, e remove os *outliers*, respectivamente, sem que ocorram perdas na qualidade das informações. Por fim, foram implementados os algoritmos Máquina de Vetor de Suporte e *k-means*.

4 RESULTADOS E DISCUSSÕES

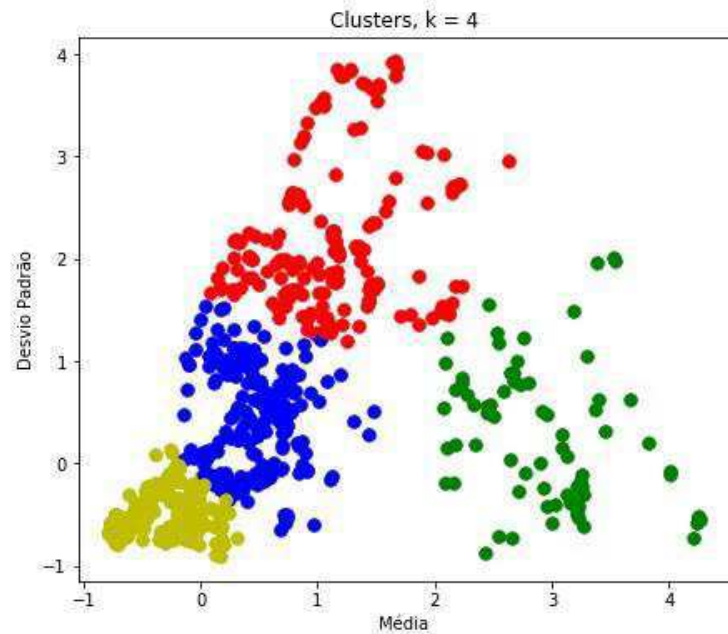
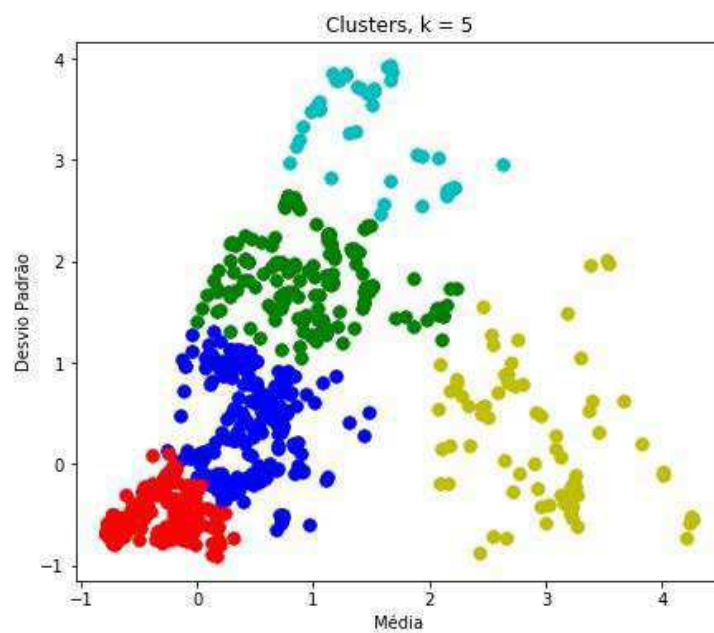
4.1 Análise de *Clustering* e Identificação de Padrões

Foi observado, como ilustrado na Figura 15, através da implementação do método Elbow que a partir de $k = 4$ não há mais uma variação significativa na redução dos erros quadráticos, com o aumento do número de clusters. Conclui-se então que as informações de consumo de energia podem ser agrupadas de forma adequada no intervalo $k \geq 4$.

Figura 15 - Curva de decrescimento da soma dos erros quadráticos em função do incremento de k .



Para a execução do algoritmo *k-means*, foram considerados valores de $k = 4$ e 5 . Os clusters obtidos para cada um são representados nas Figuras 16 e 17, respectivamente, em cores diferentes.

Figura 16 - Clusters obtidos para $k = 4$.Figura 17 - Clusters obtidos para $k = 5$.

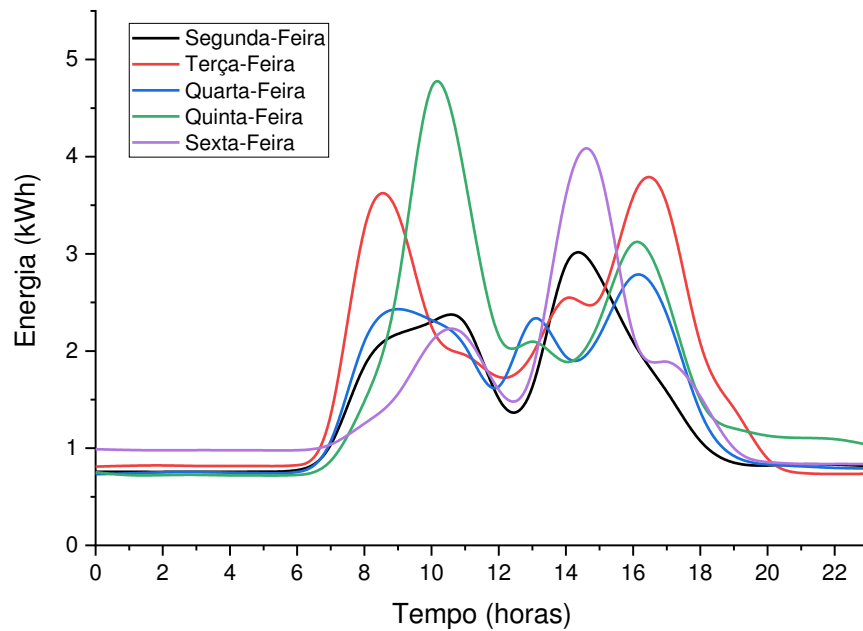
Observa-se que para $k = 5$ há uma melhor segmentação nos dados. Podemos identificar grupos com comportamentos de consumo distintos em relação aos dias e horários analisados. O cluster 2 (em vermelho) possui um comportamento estável (não possuindo picos e nem variações), representando o intervalo de tempo que vai das 19:00 às 8:00, da manhã seguinte, de todos os dias da semana. O cluster 0 (em azul) representa o consumo

médio-baixo, com poucas variações, compreendendo os horários de almoço (12:00 às 14:00) de toda semana e os fins de tarde e início das noites das segundas e sextas-feiras. Os picos observados das 8:00 às 11:00 das terças e quintas-feiras e das 14:00 às 17:00 das terças e quartas-feiras, são representados pelo cluster 4 (em amarelo). Os clusters 1 e 3 (em verde e ciano, respectivamente) compreendem as maiores variações de consumo, observadas nas manhãs das segundas, quartas e sextas-feiras (intervalo das 8:00 às 11:00 não apresentou padrão bem definido) e nas tardes das segundas, quintas e sextas-feiras. Vale salientar que o intervalo das 11:00 às 12:00 (todos os dias) apresentou um alto índice de variação de consumo (padrão também não bem definido). A descrição acima pode ser melhor observada na Tabela 2, e pode ser validada com as curvas típicas de carga apresentadas na Figura 18.

Tabela 2 - Padrões de carga por dia da semana.

Clusters	0		1	2	3	4	
Dias	Todos os dias	Segundas e Sextas	Segundas, Quintas e Sextas	Todos os dias	Todos os dias	Terças e Quartas	Terças e Quintas
Horários	12:00 às 14:00	17:00 às 19:00	14:00 às 17:00	19:00 às 8:00	13:00 às 14:00	14:00 às 17:00	8:00 às 11:00
Patamar de Consumo	Médio-Baixo	Médio-Baixo	Médio-Alto	Baixo	Médio-Alto	Alto	Alto

Figura 18 - Curvas de carga por dia da semana.



4.2 Reconhecimento de Padrões

Por ser um algoritmo de classificação supervisionada, a máquina de vetor de suporte depende do conhecimento prévio das classes de suas entradas. A SVM foi então implementada utilizando a função *kernel* do tipo RBF, capaz de boa generalização e processamento de dados que não são linearmente separáveis, com base nos agrupamentos realizados de antemão pelo *k-means*.

A construção do modelo computacional seguiu o padrão definido para algoritmos de classificação supervisionada (COSTA et al., 2011): seleciona-se uma base de dados, nesse caso os clusters identificados anteriormente, e esta é dividida em duas partes, uma parcela de treinamento (cerca de 70 %) e uma de teste (os 30 % restantes). Logo após, segue-se o script definido pelo autor (ver APÊNDICE B) para a execução.

4.2.1 Avaliação do Classificador

A avaliação da eficiência da classificação de cada cluster foi determinada pela “Matriz de Confusão” ou “Matriz de Erro” e “Acurácia Global”, como ocorre de maneira semelhante em CONGALTON e GREEN, 2009). A matriz de erro é definida da seguinte forma (Figura 19):

Figura 19 - Matriz de Confusão.

		Classe prevista	
		+	-
Classe real	+	TP True Positives	FN False Negatives Type II error
	-	FP False Positives Type I error	TN True Negatives

Fonte: Disponível em: <https://stanford.edu/~shervine/l/pt/teaching/cs-229/dicas-truques-aprendizado-maquina>.

A acurácia global (AG) foi calculada pela soma dos valores que correspondem a diagonal principal (n_{ii} = número de classificações corretas), pelo número total de amostras N , como segue o equacionamento:

$$AG = \frac{\sum_{i=1}^m n_{ii}}{N} \quad (10)$$

Sendo m , o número de classes presentes na matriz.

Desse modo, obteve-se a seguinte matriz de erro, para um total de 432 amostras (Tabela 3):

Tabela 3 - Matriz de Erro de Base de Treinamento.

SVM		Classe Prevista				
		0	1	2	3	4
Classe Real	0	30	2	3	6	7
	1	5	28	0	1	5
	2	2	0	243	1	0
	3	6	1	1	24	4
	4	7	4	1	6	45

Calculando-se a AG vide Equação (10), obtém-se uma eficiência nos acertos de 85,66 %. Para validar esse resultado, o modelo é confrontado com novas bases de dados (ver APÊNDICE B) (medições por hora de uma segunda e uma quarta-feira), referentes ao bloco CJ, obtendo-se um desempenho de 79,16 % e 75 %, respectivamente (matrizes de erro ilustradas na Figura 20). Nota-se um desempenho médio, devido a alta variabilidade nos dados, que acarreta em uma dificuldade significativa de se obter um padrão bem definido. Como apresentado anteriormente, o cluster 2, representando o intervalo medido das 19:00 às 8:00 em todos os dias, possui uma definição comportamental sólida, viabilizando então o gerenciamento de consumo de energia elétrica de forma eficiente, podendo-se identificar facilmente, por exemplo, comportamentos anômalos durante esses horários.

Figura 20 - Matrizes de erro para as classificações de consumo de uma segunda-feira (à esquerda) e de uma quarta-feira (à direita).

S	0	1	2	3	4		Q	0	1	2	3	4
0	1	2	0	0	0		0	0	2	1	1	0
1	0	4	0	1	0		1	0	3	0	0	0
2	1	0	14	0	0		2	0	0	14	0	0
3	0	1	0	0	0		3	1	1	0	1	0
4	0	0	0	0	0		4	0	0	0	0	0

5 CONCLUSÕES

Neste trabalho foi desenvolvida uma abordagem de identificação e reconhecimento de padrões através de análise de *clustering* e de classificação, em que foi possível extrair-se conhecimento de um grande volume de dados.

A utilização do medidor inteligente e sua integração com o Grafana, possibilitou a captura automática de informações sobre o consumo de eletricidade, armazenando-as em um servidor local, para posterior análise e manipulação.

O *k-means* é um método de agrupamento simples e eficaz. Uma vez que a iteração se estabilizou, cada ponto é atribuído ao seu centro de cluster mais próximo, minimizando a distância quadrática total. A criação dos grupos foi feita através do consumo médio, minuto a minuto, para dia útil da semana. Após a identificação dos clusters, procedeu-se ao estudo dos mesmos relacionando-os com as características de consumo.

O classificador SVM apresenta as vantagens de possuir alto poder de generalização, robustez e capacidade de lidar com dados ruidosos. O passo seguinte foi o treinamento do modelo desenvolvido com os dados de cada grupo identificado.

Ambos os algoritmos obtiveram bons resultados de agrupamento e classificação, identificando e reconhecendo bem os padrões.

O consumidor pode obter diversas vantagens ao ter acesso às análises de consumo de energia elétrica. Dentre elas, podemos destacar a observação de padrões de carga por espaço de tempo, a fim de se ter uma base de conhecimento, viabilizando uma gestão energética eficiente, nos moldes propostos pelo projeto SCIKE.

Como proposta de trabalhos futuros, está a utilização de novas abordagens de agrupamento, classificação e predição de consumo de energia elétrica. Implementação de algoritmos capazes de solucionar a questão da alta variância de dados em horários de consumo elevado. O desenvolvimento de propostas para a economia de energia e consequente redução de gastos, como por exemplo o monitoramento não invasivo e a desagregação de cargas.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALBERT, A. Smart Meter Driven Segmentation: What Your Consumption Says About You. IEEE Trans. Power Systems v.28 n. 4, pp. 4019-4030.
- BEZDEK, J., KELLER, J., KRISNAPURAM, R e PAL, N. Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. Springer. 2005.
- CONGALTON, R. and GREEN, K. Assessing the Accuracy of Remotely Sensed Data: Principles and Practices. 2nd Edition, CRC/Taylor & Francis, Boca Raton. 183 p. 2009.
- COSTA, A. L., MARTIN, C., CARVALHO, J. P., NETO, M. K. Agrupamento de Padrões de Curva de Carga Utilizando Algoritmos e Técnicas de Agrupamento como Alternativa Tarifária. Universidade Federal Tecnológica do Paraná. Curitiba. 2011.
- DANTAS, P. V. Identificação por Decomposição de Sinais de Consumo de Energia Elétrica. Dissertação de Mestrado. Universidade Federal do Amazonas. Manaus, p. 26. 2016.
- FAYYAD, U., PIATETSKY-SHAPIRO, G., SMYTH, P. The KDD Process of Extracting Useful Knowledge from Volumes of Data. Communications of the Association for Computing Machinery. V. 39. N. 11 p. 27-34. 1996.
- FERREIRA, A. M. S. Reconhecimento de Padrões e Tipificação de Perfis de Consumo: Contribuições para a Melhoria da Gestão na Distribuição de Energia Elétrica. Tese de Doutorado. Universidade Federal da Bahia. Salvador, p. 25. 2015.
- GASPAR, W. Reconhecimento de Padrões em Sistemas de Energia Elétrica através de uma Abordagem Geométrica para a Construção de Redes Neurais Artificiais. Tese de Doutorado. Universidade Federal de Juiz de Fora, p. 28. 2015.
- HOPFNER, F., KLAWONN, F., KRUSE, R. and RUNKLER, T. Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition. Jhon Wiley & Sons. 2000.
- JUPYTER NOTEBOOK. Acesso em 20 de novembro de 2018, disponível em: <https://jupyter.org/>.
- KANTARDZIC, M. Data Mining: Concepts, Models, Methods and Algorithms. 2nd ed. Wiley-IEEE Press. 2011.

KINTO, E. A. Otimização e Análise das Máquinas de Vetores de Suporte Aplicadas à Classificação de Documentos. Tese de Doutorado. Escola Politécnica da Universidade de São Paulo. São Paulo, p. 31-35. 2011.

MACHINE LEARNING. Acesso em 08 de dezembro de 2018, disponível em: <https://stanford.edu/~shervine/l/pt/teaching>.

MARIN, L. R. Descrição do Consumo de Energia de Consumidores Industriais e Comerciais Utilizando Descoberta de Conhecimento em Base de Dados. Dissertação de Mestrado. Universidade do Vale do Itajaí. Itajaí, p. 50-58. 2014.

MATPLOTLIB. Acesso em 20 de novembro de 2018, disponível em: <https://matplotlib.org/>.

MONZANI, R. C. Monitoramento Não-Invasivo e Identificação do Consumo de Energia Individual de Cargas Residenciais. Tese de Doutorado. Universidade Estadual de Campinas. Campinas, p. 39. 2016.

NUMPY. Acesso em 20 de novembro de 2018, disponível em: <http://www.numpy.org/>.

PANDAS. Acesso em 20 de novembro de 2018, disponível em: <https://pandas.pydata.org/>.

PRAHASTONO, I., KING, D., OZVEREN, C. and BRADLEY, D. Electricity load profile classification using Fuzzy C-means method. IEEE, p. 1-5. 2008.

SCHOLKOPF, B., PLATT, J. C., SHAWE-TAYLOR, J., SMOLA, A. J. Estimating the Support of a High Dimensional Distribution. Technical report, Microsoft Research, MSR-TR-99-87, 2002.

SCIKE, Scike - Software Cluster. Acesso em 07 de dezembro de 2018, disponível em <https://scikepb.dcx.ufpb.br/>.

SEABORN, Statistical Data Visualization. Acesso em 20 de novembro de 2018, disponível em: <https://seaborn.pydata.org/>.

SOUZA, W. A., MARAFÃO, F. P., GARCIA, F. D., PAES F. L. Gerenciamento de Consumo Baseado em Medidores Cognitivos de Energia Elétrica. SBAI 2017, Simpósio Brasileiro de Automação Inteligente. Porto Alegre, 2017.

ZHENG, J., GAO, D. W., LIN, L. Smart Meters in Smart Grid: An Overview: IEEE Green Technologies Conference p. 57–64. 2013.

APÊNDICE A

ALGORITMO K-MEANS

```
# Importação das Bibliotecas e Entrada de Dados

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

medicoes = pd.read_csv('versaofinal.csv')

# Pré Processamento

from sklearn import preprocessing

scaler = preprocessing.StandardScaler().fit(medicoes[['Media']])
medicoes['Media'] = scaler.transform(medicoes[['Media']])
scaler = preprocessing.StandardScaler().fit(medicoes[['Desvio']])
medicoes['Desvio'] = scaler.transform(medicoes[['Desvio']])
train = medicoes[['Media', 'Desvio']]

# Biblioteca KMeans

from sklearn.cluster import KMeans

# Implementação do Método do Cotovelo

wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'random')
    kmeans.fit(train)
    print (i, kmeans.inertia_)
    wcss.append(kmeans.inertia_)

plt.plot(range(1, 11), wcss)
plt.title('Método Elbow')
```

```

plt.xlabel('Número de Clusters')
plt.ylabel('Soma dos Erros Quadráticos')
plt.show()

# Implementação do k-means
kmeans = KMeans(n_clusters = 5, init = 'random')
kmeans.fit(train)
centroides = kmeans.cluster_centers_
rotulos = kmeans.labels_
cores = ['g.', 'r.', 'b.', 'y.', 'c.']
vector = medicoes.values

# Visualização de Resultados
fig, axes1 = plt.subplots(figsize=(7,6))
for j in range(len(medicoes['Desvio'])):
    axes.plot(vector[j][0], vector[j][1], cores[rotulos[j]], markersize = 15)
axes1.set_xlabel('Média')
axes1.set_ylabel('Desvio Padrão')
axes1.set_title('Clusters, k = 4')

fig, axes2 = plt.subplots(figsize=(7,6))
for n in range(len(medicoes['Desvio'])):
    axes.plot(vector[n][0], vector[n][1], cores[rotulos[n]], markersize = 15)
axes2.set_xlabel('Média')
axes2.set_ylabel('Desvio Padrão')
axes2.set_title('Clusters, k = 5')

# Exporta Resultados
G = pd.DataFrame(group)
G.to_csv("group.csv")

```

ALGORITMO SVM

Importa Bibliotecas

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

%matplotlib inline

data = pd.read_csv('svm.csv')

Pré Processamento

from sklearn import preprocessing

scaler = preprocessing.StandardScaler().fit(data[['Media']])

data['Media'] = scaler.transform(data[['Media']])

scaler = preprocessing.StandardScaler().fit(data[['Desvio']])

data['Desvio'] = scaler.transform(data[['Desvio']])

Construção do Modelo

from sklearn.model_selection import train_test_split

X = data[['Media', 'Desvio']]

Y = data['Classe']

train_test_split(X, Y, test_size = 0.3)

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.3)

Implementação da SVM

from sklearn.svm import SVC

model = SVC()

model.fit(X_train, Y_train)

pred = model.predict(X_test)

from sklearn.metrics import classification_report, confusion_matrix

```
# Visualização de Resultados  
print(classification_report(Y_test, pred))  
print('\n')  
print(confusion_matrix(Y_test, pred))  
  
# Validação  
newpattern = pd.read_csv('predict.csv')  
newpattern.head()  
validation = newpattern.values  
print(model.predict(validation))
```

APÊNDICE B

Segunda-Feira					
Hora	Med. Cons	M	D	Patamar k = 5	Patamar k = 4
0	0,75756967	-0,74673	-0,66651	2	2
1	0,75625817	-0,74802	-0,6666	2	2
2	0,75576717	-0,74849	-0,66723	2	2
3	0,75767592	-0,74663	-0,66404	2	2
4	0,75518992	-0,74906	-0,66678	2	2
5	0,75944425	-0,7449	-0,6677	2	2
6	0,75601825	-0,74825	-0,66578	2	2
7	0,86596767	-0,64091	-0,57347	2	2
8	2,00284533	0,46899	1,3033	1	0
9	2,19640008	0,65795	0,86929	1	0
10	2,26704808	0,72692	1,38385	1	1
11	2,533224	0,98678	1,35413	1	1
12	1,30837642	-0,209	-0,65815	2	2
13	1,29373933	-0,22329	-0,44376	0	2
14	3,30381	1,73908	1,32481	3	1
15	2,81534825	1,26221	1,2044	1	1
16	2,03536325	0,50074	-0,19568	0	0
17	1,61317533	0,08857	-0,32795	0	2
18	1,00363042	-0,50651	-0,56209	2	2
19	0,82126825	-0,68455	-0,61968	2	2
20	0,81776325	-0,68797	-0,62615	2	2
21	0,83824242	-0,66798	-0,59225	2	2
22	0,83242225	-0,67366	-0,60436	2	2
23	0,81633658	-0,68936	-0,6379	2	2

Quarta					
Hora	Med. Cons	M	D	Patamar k = 5	Patamar k = 4
0	0,7323565	-0,77135	-0,59353	2	2
1	0,73953642	-0,76434	-0,60035	2	2
2	0,75367558	-0,75054	-0,6109	2	2
3	0,75916417	-0,74518	-0,59142	2	2
4	0,74562167	-0,7584	-0,60933	2	2
5	0,74428758	-0,7597	-0,60724	2	2
6	0,73371017	-0,77003	-0,59722	2	2
7	0,85026708	-0,65624	-0,61135	2	2
8	2,30903383	0,76791	2,81506	3	1
9	2,49094267	0,94551	2,50994	3	1
10	2,30970425	0,76857	0,29836	1	0
11	2,18739692	0,64916	0,48138	1	0
12	1,21448525	-0,30066	-0,24678	0	2
13	2,75578533	1,20406	0,86966	1	0
14	1,70986992	0,18297	-0,6371	0	2
15	2,06463875	0,52932	-0,09882	0	0
16	3,01339033	1,45556	1,65036	3	1
17	2,50714258	0,96132	1,27451	1	1
18	1,22155	-0,29376	0,17498	0	2
19	0,86716525	-0,63974	-0,63991	2	2
20	0,8298265	-0,67619	-0,66752	2	2
21	0,81328358	-0,69234	-0,7423	2	2
22	0,79520025	-0,71	-0,77861	2	2
23	0,7906915	-0,7144	-0,77943	2	2