UNIVERSIDADE FEDERAL DO AMAZONAS - UFAM

FACULDADE DE TECNOLOGIA

BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

Aprendizagem de máquina como ferramenta de planejamento energético: estudo de caso aplicado a comunidades não eletrificadas do baixo Rio Negro no Amazonas

Luís Henrique Raheem Simões

Manaus - AM

Novembro de 2023

Luís Henrique Raheem Simões

Aprendizagem de máquina como ferramenta de planejamento energético: estudo de caso aplicado a comunidades não eletrificadas do baixo Rio Negro no Amazonas

Monografia de Graduação apresentada a Faculdade de Tecnologia da Universidade Federal do Amazonas como requisito parcial para a obtenção do grau de Bacharel em Engenharia a Computação

Orientador

Alessandro Bezerra Trindade, Dr.

Universidade Federal do Amazonas - UFAM

Faculdade de Tecnologia

Manaus - AM

Novembro de 2023

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

Simões, Luís Henrique Raheem
S593a Aprendizagem de máquina como ferramenta de planejamento energético : estudo de caso aplicado a comunidades não eletrificadas do baixo Rio Negro no Amazonas / Luís Henrique Raheem Simões . 2023
100 f.: il. color; 31 cm.

Orientador: Alessandro Bezerra Trindade
TCC de Graduação (Engenharia da Computação) - Universidade Federal do Amazonas.

1. Aprendizagem de máquina supervisionado. 2. Mineração de dados. 3. Planejamento energético. 4. Comunidades ribeirinhas. 5. Ciência de dados. I. Trindade, Alessandro Bezerra. II. Universidade Federal do Amazonas III. Título

Monografia de Graduação sob o título *Aprendizagem de máquina como ferramenta de planejamento energético: estudo de caso aplicado a comunidades não eletrificadas do baixo Rio Negro no Amazonas* apresentada por Luís Henrique Raheem Simões e aceita pela Faculdade de Tecnologia da Universidade Federal do Amazonas, sendo aprovada por todos os membros da banca examinadora abaixo especificada:

Prof. Dr. Alessandro Bezerra Trindade
Universidade Federal do Amazonas - UFAM
Faculdade de Tecnologia
Orientador

Prof. Dr. Francisco de Assis Pereira Januário
Universidade Federal do Amazonas - UFAM
Faculdade de Tecnologia
Examinador

Prof. Dr. Ozenir Farah da Rocha Dias
Universidade Federal do Amazonas - UFAM
Faculdade de Tecnologia
Examinador

Manaus - AM, 08 de Novembro de 2023.

A uma jornada que começou a muitos anos.

Agradecimentos

Agradeço à minha família, pelo apoio e incentivo incondicional.

Agradeço ao meu orientador, Alessandro Trindade, pela orientação, paciência e dedicação. Sempre esteve disponível para me ajudar, mesmo fora do horário de trabalho. Suas orientações foram essenciais para o desenvolvimento deste trabalho, não esquecendo da possibilidade que ele me proporcionou para escrever um artigo que resultou neste trabalho.

Agradeço ao meu colega, Samuel Gunnar, que escreveu comigo o artigo que resultou no desenvolvimento deste trabalho.

Agradeço ao meu grande amigo, Nader Hauache, pelos conselhos que me deu durante o desenvolvimento deste trabalho.

Agradeço ao meu querido amigo, André Girão, que compartilhou comigo os seus conhecimentos sobre Inteligência Artificial.

Agradeço a minha colega, Caroline Braz, pela ajuda na confecção de imagens neste trabalho.

Agradeço minha antiga terapeuta, Virgínia de Souza, que me ensinou a manter a saúde mental em tempos de grande dificuldade.

Por fim, agradeço a todos aqueles que contribuíram para o meu crescimento acadêmico e pessoal ao longo desta árdua jornada.

Com sincera gratidão,

Luís Henrique Raheem Simões

O trabalho deve ser realizado, não o fruto do trabalho.

Krishna

Aprendizagem de máquina como ferramenta de planejamento energético: estudo de caso aplicado a comunidades não eletrificadas do baixo Rio Negro no Amazonas

Autor: Luís Henrique Raheem Simões

Orientador: Alessandro Bezerra Trindade, Dr.

Resumo

Este trabalho propõe a utilização da ciência de dados como ferramenta de planejamento energético e poder auxiliar no dimensionamento de sistemas elétricos de comunidades isoladas na região amazônica. Embora seja uma região imensa, a bacia amazônica ainda apresenta um desafio muito grande para se levar energia elétrica, principalmente pela falta de dados detalhados das comunidades e das suas necessidades energéticas. E, sem um dimensionamento correto, os custos envolvidos, sejam de instalação quanto de operação dos sistemas de eletrificação, podem causar prejuízo para a concessionária de energia ou ainda insatisfação dos usuários pelo não atendimento às suas necessidades. O método aqui proposto é baseado em aprendizagem de máquina supervisionado, utilizando vários classificadores para prever a potência instalada e a energia consumida de casas de uma comunidade ribeirinha. Os dados utilizados para o treinamento do método foram coletados de 14 comunidades não eletrificadas no baixo Rio Negro, no Amazonas. Os questionários coletaram informações sobre as características socioeconômicas das famílias, como estrutura familiar, nível de educação, atividades produtivas e hábitos de consumo de energia. Os resultados do método mostraram que é possível prever a potência instalada e a energia consumida com boa precisão. A potência instalada

foi prevista com uma acurácia de 79,2% e a energia consumida foi prevista com uma acurácia de 68,5%. Este método pode ser utilizado para apoiar a tomada de decisão na eletrificação de comunidades isoladas. Pode ajudar a estimar os custos de um sistema de eletrificação, bem como a identificar as comunidades que têm maior necessidade energética.

Palavras-chave: Aprendizagem de Máquina Supervisionado, Planejamento Energético, Comunidades Ribeirinhas.

Aprendizagem de máquina como ferramenta de planejamento energético: estudo de caso aplicado a comunidades não eletrificadas do baixo Rio Negro no Amazonas

Autor: Luís Henrique Raheem Simões

Orientador: Alessandro Bezerra Trindade, Dr.

Abstract

This work proposes the use of data science as an energy planning tool and can assist in the design of electrical systems in isolated communities in the Amazon region. Although it is a huge region, the Amazon basin still presents a very big challenge in providing electricity, mainly due to the lack of detailed data on communities and their energy needs. And, without correct sizing, the costs involved, whether for installation or operation of the electrification systems, can cause losses for the energy concessionaire or even dissatisfaction among users due to failure to meet their needs. The method proposed here is based on supervised machine learning, using several classifiers to predict the installed power and energy consumed of houses in a riverside community. The data used to train the method was collected from 14 non-electrified communities in the lower Rio Negro, in Amazonas. The questionnaires collected information on the socioeconomic characteristics of families, such as family structure, level of education, productive activities and energy consumption habits. The results of the method showed that it is possible to predict the installed power and energy consumed with good accuracy. The installed power was predicted with an accuracy of 79.2% and the energy consumed was predicted with an accuracy of 68,5%. This method can be used to support

decision-making in the electrification of isolated communities. It can help estimate electrification costs as well as identify communities that have the greatest energetic need.

Keywords: Supervised Machine Learning, Energy Planning, Riverine Communities.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AD Árvore de Decisões

AM Aprendizagem de Máquina

AMS Aprendizagem de Máquina Supervisionada

APA Área de Proteção Ambiental

BA *Bootstrap Aggregating*

BDP Banco de Dados Padrão

CD Ciência de Dados

CSV *comma-separated-values*

CV *Cross Validation*

DC *Data Cleaning*

DT *Decision Tree*

GC *Google Colaboratory*

IA Inteligência Artificial

IBL *Instance-Based Learning*

IEMA Instituto de Energia e Meio Ambiente

KNN *K-Nearest Neighbor*

LL *Lazy Learning*

LP Linguagem de Programação

MD Mineração de Dados

ML *Machine Learning*

NB *Naive Bayes*

OR *One Rule*

PANDAS *Python Data Analysis Library*

RDS Reserva de Desenvolvimento Sustentável

RS *Random Subspace*

Scikit-learn *Machine Learning in Python*

TB Teorema de Bayes

WEKA *Waikato Environment for Knowledge Analysis*

SUMÁRIO

1

INTRODUÇÃO

A Mineração de Dados e Inteligência Artificial são duas áreas inter-relacionadas da Ciência da Computação, e a Aprendizagem de Máquina é uma subárea a Inteligência Artificial. Juntas, possibilitam o acesso a coleções de métodos que foram criados a partir de modelos matemáticos e teoria estatística permitindo a automatização de diversas tarefas baseando-se na descoberta sistemática dos padrões em conjuntos de dados disponíveis para testes ou em experiências do passado que foram registradas e convertidas em dados (ALPADIN, 2020). Desenvolvedores tem usado a Ciência de Dados para executar tarefas de maneira eficiente que outrora eram feitas manualmente, como conectar-se com clientes, identificar padrões e resolver problemas (ORACLE, 2023).

Segundo a IBM (IBM, 2023a), aprendizado supervisionado é uma subcategoria da Aprendizagem de Máquina e de Inteligência Artificial. É definida pelo uso de conjunto de dados rotulados para treinar algoritmos que classificam dados ou preveem resultados com precisão.

A partir de questionários coletados de comunidades ribeirinhas do Baixo Rio Negro é possível deduzir informações necessárias para a eletrificação das comunidades, mesmo partindo de dados preliminares incompletos ou ainda não atualizados (TRINDADE et al., 2022). Portanto, usar Aprendizagem de Máquina como pré-requisito para prever a potência instalada e energia consumida em comunidades isoladas na região amazônica a partir de questionário socieconômico é importante para o estudo de caso

de centenas de comunidades que características similares para a implementação de um projeto de eletrificação rural.

1.1 Contextualização e Definição do Problema

Existe uma riqueza de informações que podem ser extraídas dos dados através da ciência de dados, combinando várias áreas de conhecimento como Inteligência Artificial, Estatística e Aprendizagem de Máquina (MURPHY, 2012). Quando se aplica a ciência de dados em um banco de dados pode-se encontrar as tendências e tomar as melhores decisões através de previsões para aplicar um projeto ou oferecer algum serviço (NADALIN R., 2023).

A aplicação de ciência de dados em questionários coletados de comunidades ribeirinhas do baixo Rio Negro no Amazonas pode permitir a extração de informações relevantes, identificação de tendências e avaliação de correlações de atributos pesquisados nas comunidades. Esses dados são essenciais para estimar o sucesso da aplicação de projetos ou serviços e ajudar a tomar melhores decisões a partir de previsões.

Os dados coletados são de um total de 14 comunidades não eletrificadas, que foram alvo de uma pesquisa realizada entre junho e agosto de 2017. Os questionários foram realizados em cada casa, cobrindo 56 macro-questões divididas em 9 seções. O universo das 14 comunidades em 2017 era de 593 famílias, mas o número de famílias que responderam o questionário corresponde a cerca de 30% desse total.

Com esses dados coletados foi construído um dataset composto por 166 atributos e 179 instâncias. O problema consiste em realizar previsões de respostas confiáveis para comunidades similares usando ciência de dados e aprendizagem de máquina, comparando os resultados obtidos nas ferramentas WEKA e linguagem de programação Python, destacando quem obteve o melhor resultado para os atributos de potencia instalada e consumo de energia diário. Estes atributos possibilitam a realização de estudos para a eletrização de comunidades isoladas através da acurácia alcançada. O método pode fornecer uma previsão de carga para essas comunidades sem a necessidade

de ir até o local isolado, reduzindo os custos de projetos de eletrização.

1.2 Objetivos

Neste trabalho, propomos o uso de mineração de dados e aprendizagem de máquina que são subcampos da ciência de dados para prever informações de comunidades em dados detalhados coletados e que possam servir para a atividade de planejamento da eletrificação de comunidades ribeirinhas isoladas de similar características.

1.2.1 Objetivo geral

Provar que é possível usar técnicas de mineração de dados e aprendizagem de máquina em dados reais coletados de comunidades rurais não eletrificadas utilizando ferramentas de aprendizagem de máquina supervisionada, comparando os resultados obtidos e destacando qual das ferramentas obtém o melhor valor de acurácia na construção de conhecimento relevante no planejamento e eletrificação rural de comunidades sem que se tenham dados completos destas. À partir da estimativa com acurácia da potência instalada e do consumo de energia diário das casas de uma comunidade, pode-se projetar e instalar um sistema de eletrificação que universalize o acesso à energia elétrica. E, considerando-se as comunidades ribeirinhas isoladas da Amazônia, nas quais raramente se tem dados detalhados dos moradores, além do desafio logístico, um método que ajude na estimativa de parâmetros essenciais para os projetos de eletrificação é de suma importância no planejamento energético brasileiro, onde dezenas de milhares de comunidades ainda não receberam energia elétrica.

1.2.2 Objetivos específicos

1. Obter uma base de dados real de comunidades rurais da amazônia, para servir de base para as técnicas de ciência de dados.
2. Realizar o pré-processamento de dados, visando facilitar o trabalho computacional e otimizar os resultados.
3. Definir os atributos da base de dados que têm mais correlação entre si.
4. Testar a base de dados em função dos atributos correlacionados em ferramenta *open-source* (WEKA).
5. Desenvolver em *Python* uma solução que teste vários algoritmos de aprendizagem de máquina visando a obtenção da melhor acurácia.
6. Comparar os resultados para definir o melhor método de estimativa de potência instalada e da energia elétrica de cada casa de uma comunidade.

1.3 Organização do Documento

Este documento está organizado da seguinte forma:

Capítulo 1 - INTRODUÇÃO: Este capítulo fornece uma visão geral do trabalho desenvolvido, apresentando os objetivos e justificativas para o trabalho.

Capítulo 2 - FUNDAMENTAÇÃO TEÓRICA: Neste capítulo, é realizada uma revisão da literatura, explorando estudos e trabalhos existentes que servem como base teórica para o desenvolvimento trabalho.

Capítulo 3 - METODOLOGIA: Este capítulo apresenta os métodos utilizados para testes na base de dados. São descritos todos os algoritmos utilizados e a acurácia de cada um deles com diferentes estratégias de previsão.

Capítulo 4 - RESULTADOS E DISCUSSÕES: Neste capítulo, é apresentada a comparação dos métodos utilizados, a melhor ferramenta e método, e os resultados obtidos.

Capítulo 5 - CONCLUSÃO: Este capítulo apresenta as conclusões obtidas com base nos resultados e discussões realizadas anteriormente. Também são discutidos os objetivos alcançados e são apresentadas sugestões para trabalhos futuros, com o intuito de aprimorar os algoritmos e explorar novas possibilidades de pesquisa.

2

FUNDAMENTAÇÃO TEÓRICA

Este capítulo introduz os dados sobre a não eletrificação da região amazônica, desafios logísticos e dimensionamento de carga, a falta de literatura sobre planejamento energético na amazônia. Também apresenta os conceitos e ferramentas utilizadas na concepção deste trabalho: A Aprendizagem de Máquina Supervisionada, os conceitos que definem os dados, a informação, o conhecimento e a sabedoria, a limpeza de dados e a importância para se trabalhar com a Aprendizagem de Máquina Supervisionada, a ferramenta WEKA e seus métodos, *Google Colaboratory* e a facilidade de trabalhar com Aprendizagem de Máquina Supervisionada.

2.1 Dados de não eletrificação na região amazônica

Segundo dados do Instituto de Energia e Meio Ambiente (IEMA, 2022), havia cerca de 70 mil famílias sem acesso à energia elétrica na região amazônica no ano de 2022. Isso representa cerca de 280 mil pessoas que vivem sem energia elétrica. Esse número representa um desafio significativo para o desenvolvimento da região amazônica. A energia elétrica é essencial para o acesso a serviços básicos, como educação, saúde e comunicação. Também é importante para o desenvolvimento econômico da região, que é dependente de atividades como agricultura, pesca e turismo. Infelizmente, a instalação de rede elétrica em regiões rurais da Amazônia enfrenta uma série de desafios logísticos.

Dentre os principais desafios, destacam-se:

- Acesso às comunidades: muitas comunidades rurais da Amazônia estão localizadas em áreas de difícil acesso, o que dificulta o transporte de materiais e equipamentos.
- Custo de transporte: o transporte de materiais e equipamentos para áreas remotas da Amazônia é caro, o que pode aumentar o custo final do projeto de eletrificação.
- Condições climáticas: as condições climáticas da Amazônia, como chuvas intensas e inundações, podem dificultar ou impedir as obras de construção e manutenção da rede elétrica.

Para dimensionar a rede elétrica de forma adequada, é necessário conhecer a demanda de energia elétrica da comunidade. Uma das formas de estimar a demanda de energia elétrica é utilizar o método da previsão da potência instalada e consumo diário de energia. Esse método utiliza dados cadastrais, escolaridade, renda e tipo de moradia. O método pode ser uma alternativa viável para estimar a demanda de energia elétrica em comunidades rurais da Amazônia, pois não é necessário realizar levantamentos de campo.

2.2 Literatura sobre planejamento energético e o uso de Aprendizagem de Máquina

Não há literatura sobre planejamento energético similar ao estudo de caso proposto neste trabalho. A literatura encontrada geralmente se concentra em sistemas elétricos de potência para prever carga elétricas ou resiliência de sistemas de potência em geral, não para regiões isoladas e muito menos para a região amazônica. Alguns dos trabalhos relacionando a Aprendizagem de Máquina e planejamento energético são:

Artigo: Aplicações de Aprendizagem de Máquina ML em sistemas elétricos

Segundo Wang (WANG, 2020), a Aprendizagem de Máquina tem sido cada vez mais adotada em pesquisas e aplicações em sistemas elétricos. O artigo cita como exemplos o uso da AM para previsão de carga e detecção de falhas, que são tópicos bem estudados. O autor também menciona que a AM está ganhando popularidade para aplicações como controle de tensão e análise de distribuição.

Artigo: O aprendizado de máquina para sistemas modernos de distribuição de energia: Progresso e perspectivas

Segundo Markovic *et al.* (MARKOVIć; BOSSART; HODGE, 2023), a aplicação da Aprendizagem de Máquina em sistemas de energia e energia está sendo pesquisada a uma taxa surpreendente, resultando em um número significativo de adições recentes à literatura. À medida que a infraestrutura dos sistemas de energia elétrica evolui, também aumenta o interesse na implantação de técnicas de AM. No entanto, apesar do crescente interesse, o número limitado de aplicações no mundo real relatadas sugere que a lacuna entre a pesquisa e a prática ainda não foi totalmente preenchida.

Artigo: A aplicação do aprendizado de máquina para aprimorar a resiliência do sistema de energia

Segundo Xie *et al.* (XIE; ALVAREZ-FERNANDEZ; SUN, 2020), é um desafio urgente integrar a tecnologia avançada de aprendizado de máquina e uma grande quantidade de dados em tempo real de sistemas de medição de ampla área e dispositivos eletrônicos inteligentes, a fim de aprimorar efetivamente a resiliência do sistema de energia e garantir a operação confiável e segura dos sistemas de energia. O artigo visa revisar sistematicamente a aplicação existente de métodos de aprendizado de máquina na melhoria da resiliência do sistema de energia, expandir o interesse de pesquisadores e acadêmicos sobre esse tópico e promover conjuntamente a aplicação da inteligência

artificial no campo de sistemas de energia.

2.3 Mineração de dados e aprendizagem de máquina supervisionada

A mineração de dados e a aprendizagem de máquina são duas áreas distintas, mas correlacionadas da Ciência de Dados.

A mineração de dados é um processo de descoberta de padrões e relacionamentos nos dados. É uma área multidisciplinar que se baseia em técnicas de estatística, aprendizagem de máquina, inteligência artificial e visualização de dados. A mineração de dados pode ser usada para uma variedade de propósitos, incluindo:

- **Reconhecimento de padrões:** A mineração de dados pode ser usada para identificar padrões em dados, como tendências ou comportamentos.
- **Classificação:** A mineração de dados pode ser usada para classificar dados em categorias.
- **Regressão:** A mineração de dados pode ser usada para prever valores futuros.
- **Agrupamento:** A mineração de dados pode ser usada para agrupar dados em grupos semelhantes.

A aprendizagem de máquina é o campo da ciência de dados que se concentra na construção de algoritmos que podem aprender com dados e melhorar seu desempenho ao longo do tempo. A Aprendizagem de Máquina Supervisionada treina um modelo com comandos rotulados, ou seja, dados que incluem um rótulo que identifica uma saída correta. O modelo aprende a mapear entradas para saídas com base nesses dados rotulados (IBM, 2023b).

Existem dois tipos de aprendizagem supervisionada:

- **Classificação:** O modelo aprende a classificar dados em categorias usando um algoritmo para separar com precisão os dados de teste em categorias específicas.

Os algoritmos mais comuns são classificadores lineares, maquinas de vetores de suporte, árvore de decisão, k-vizinhos mais próximos entre outros.

- **Regressão:** O modelo prevê um valor numérico entendendo a relação entre variáveis dependentes e independentes. Os algoritmos mais comuns são os de regressão linear e regressão logística.

O processo de aprendizagem de máquina supervisionada pode ser dividido em três etapas:

1. **Coleção dos dados:** É a etapa onde os dados são rotulados e coletados.
2. **Treinamento do modelo:** É a etapa onde o modelo rotulado é treinado com os dados rotulados e aprende a mapear entradas para saídas com base nos dados rotulados.
3. **Teste do modelo:** É a etapa onde o modelo é testado com novos dados que não foram usados para treinar o modelo.

A Aprendizagem de Máquina Supervisionada é simples de implementar e é eficaz para uma gama de aplicações, mas requer dados rotulados para treinar modelos (PONCE et al., 2021). Dependendo de como os dados foram rotulados, o desempenho do modelo pode ser afetado, portanto é preciso saber escolher os dados que serão treinados.

2.4 O uso da pirâmide da sabedoria em ciência de dados

O modelo da pirâmide da sabedoria é amplamente utilizado nas áreas de computação em Ciência de Dados e Aprendizagem de Máquina Figura 1.

Este modelo ilustra a relação entre dados, informação, conhecimento e sabedoria. Os dados são a base da pirâmide e são essenciais para a criação da informação. A informação é o fundamento para o conhecimento. O conhecimento é o que permite chegarmos na sabedoria, o que resulta em decisões mais assertivas.

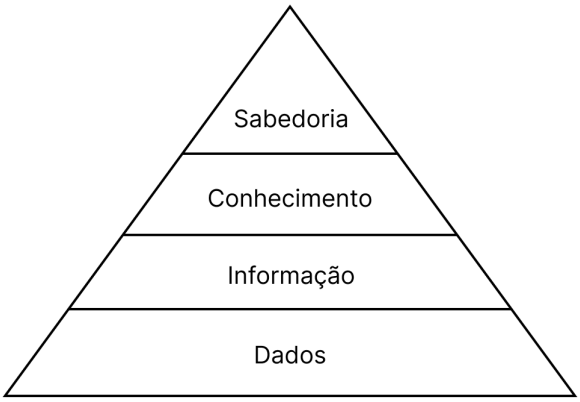

Figura 1 – Pirâmide da sabedoria

2.4.1 Dados

Dados são representações simbólicas da informação, isto é, factos ou entidades do mundo, retratado por um valor simbólico (MüLLER; FREYTAG, 2015). Os dados podem ser apresentados como um conjunto de palavras, números, símbolos, arquivos salvos em banco de dados e outros produtos de observação e medição, caracterizados pela falta de sentido e interpretação, mas que podem ser transformados em dados úteis quando qualificados.

Quanto a estrutura dos dados, elas podem se apresentar das seguintes formas:

- **Não estruturadas:** Dados sem estrutura predefinida, sem esquema e sem regras.
- **Semiestruturadas:** Apresentam uma estrutura heterogênea e predefinida, mas irregular e podem conter ou não um esquema de dados.
- **Estruturados:** São dados organizados em uma estrutura predefinida, regular e rígida, possuindo esquemas de dados fechados.

Quanto ao tempo dos dados, são observados da seguinte maneira:

- **t0: não estruturado** Os dados estão disponíveis no ambiente, já foram notados, mas não receberam tratamento e encontram-se brutos, crus.
- **t1: semiestruturado** Os dados começaram a receber tratamento ou receberam um tratamento irregular, incompleto.

- **t2: estruturados** Os dados foram tratados.

Dados semiestruturados (t1) e estruturados(2) não são informação, pois foram apenas organizados e estão prontos ou quase prontos para receberem um significado ou contexto. Somente quando são colocados em contextos é que os dados passam a ser informação (RIBEIRO, 2010).

Os dados devem estar disponíveis no sistema do computador ou em nuvem na forma de tabela. Dessa maneira os dados podem ser acessados quase que universalmente. Numa tabela, cada linha/registro representa uma instância/entidade e cada um de seus valores/coluna representam uma variável/propriedade da entidade.

2.4.2 Informação

Informação é a interpretação dos dados. A partir da interpretação, os dados recebem significação, utilidade, processamento, contexto ou interpretação. Pode-se dizer que estes dados tratados e interpretados passam uma mensagem dentro de um contexto real. A expressão da informação para uma futura análise de dados deve ser descrita ou representada de uma maneira física, como um sinal, texto ou comunicação para que seja atribuível a objetos como dados, textos e documentos. E por fim a informação pode ser convertida em dado por meio de captura e armazenamento (LIEW, 2013).

2.4.3 Conhecimento

Conhecimento se constrói através da habilidade de analisar as informações. Quando as informações são integradas e processadas são produzidas reflexões e conclusões que somente a informação não é capaz de gerar. Na verdade, em razão de sua natureza intangível e difusa, definir conhecimento é precisamente difícil (BHATT, 2002). Entretanto, para fins didáticos, pode-se definir o conhecimento em dois tipos:

- **Conhecimento tácito:** É o conhecimento gerado através de experiências práticas e empíricas. Pode ser internalizado, dificilmente transmitido, compartilhado e formalizado.
- **Conhecimento explícito:** É o conhecimento baseado em teorias. É codificado, formal, transmitido de forma sequencial e estruturada, podendo ser facilmente transmitido ou explicado, geralmente por meio da escrita.

Neste trabalho, através da manipulação das informações com ferramentas específicas chega-se ao conhecimento.

2.4.4 Sabedoria

Sabedoria pode ser compreendida como a capacidade de tomar decisões corretas a partir do novo conhecimento adquirido. A sabedoria é voltada a ação da mesma forma que a informação e o conhecimento. Mas diferentemente dos demais, parece relacionada a algo mais integrador, compreendendo o interesse em escolher um comportamento apropriado à situação, por meio de análise e síntese do conhecimento, para obtenção de um resultado positivo em escala global (HOPPE et al., 2011). A sabedoria se refere a fazer as escolhas certas com eficácia e eficiência.

2.5 Pré-processamento dos dados

O pré-processamento de dados é uma etapa fundamental do processo de mineração de dados. Ele garante que os dados estejam em um formato adequado para serem analisados pelos algoritmos de mineração (CHANDRASHEKAR; RAMAKRISHNAN, 2014).

As técnicas de pré-processamento de dados podem ser divididas em três categorias principais:

- **Limpeza de dados:** remoção de anomalias dos dados, como valores ausentes, inválidos ou incoerentes.
- **Transformação de dados:** modificação da estrutura ou do formato dos dados para torná-los mais adequados para análise.
- **Redução de dimensionalidade:** redução do número de atributos dos dados para melhorar o desempenho dos algoritmos de mineração.

2.6 Limpeza de dados

Data Cleaning, ou Limpeza de Dados - também conhecida *Scrubbing* - é um conjunto de técnicas utilizadas para detectar e remover anomalias em bases de dados (RAHM; DO, 2000). A limpeza dos dados é o primeiro passo e mais importante quando trabalha-se com ciência de dados, pois até o algoritmo mais sofisticado pode retornar respostas duvidosas se os dados não estiverem devidamente limpos.

De acordo com Vasco (VASCO, 2013), as anomalias existentes em bancos de dados podem ser divididas em três categorias que são as anomalias de sintaxe, de semântica e de cobertura.

2.6.1 Anomalias de sintaxe

Segundo Müller e Freytag (MüLLER; FREYTAG, 2015), as anomalias de sintaxe condizem ao formato e valores adotados para a representação do dado. Podem conter erros lexicais, erros no formato do domínio e irregularidades. Essas anomalias podem ser classificadas da seguinte forma:

- **Erros lexais:** Acontece quando um registro tem mais ou menos elementos do que deveria ter. Um exemplo de erro lexal pode ser visto na Tabela 1:

Tabela 1 – Dados com erros lexais

Gênero	Potência instalada	Consumo de energia diário
F	Até 1000W	De 1KWh até 4KWh
F	F	
M	De 1000W até 2000W	

- **Erros no formato do domínio:** Acontece quando o valor atribuído a um atributo não corresponde ao formato que foi pré-estabelecido. Por exemplo, se determinamos uma variável LOCAL com domínio específico dado por "Cidade - Estado"a entrada "Manaus Amazonas"está incorreta porque falta o hífen.

Esses dois itens são violações no formato geral e por este motivo podem ser denominados **erros de sintaxe**.

2.6.2 Anomalias de semântica

As anomalias de semântica condizem ao não entendimento do dado registrado. Podem conter violações das restrições de integridade, contradições, duplicidade de registros e registros inválidos.

- **Violações das restrições de integridade:** Acontecem quando um registro ou um conjunto não satisfazem uma ou mais restrições de integridade. Por exemplo, uma variável IDADE cuja variável precisa ser maior que zero (IDADE > 0).

- **Contradições:** Acontecem quando valores dentro do banco de dados violam algum tipo de dependência entre valores como a violação de dependências funcionais que podem ser representadas como restrições de integridade ou duplicados com valor incorreto. Por exemplo, têm-se as variáveis GÊNERO E GRAVIDEZ para dados que representam que uma pessoa do gênero masculino está grávida.

- **Duplicados:** Acontecem quando dois ou mais registros representam a mesma entidade dentro de um conjunto de dados. Quanto aos duplicados inexatos, eles representam a mesma entidade, mas com valores diferentes para todas ou algumas das suas propriedades. Dados desse tipo podem dificultar a detecção e correção de duplicados.

- **Registros Inválidos:** Acontecem em dados que não apresentam nenhuma das anomalias já mencionadas, entretanto não representam uma uma entidade válida. É considerada a classe mais complexa de se tratar, pois não violam qualquer regra ou restrição são complicadas de corrigir e difíceis deletar.

2.6.3 Anomalias de cobertura

As anomalias de cobertura condizem à ausência de informação quando esta for uma premissa do dado. Podem conter valores omissos e registros omissos.

- **Valores omissos:** Acontecem quando há ausência de informação sobre alguma entidade. Caso o atributo seja *NULL* a anomalia pode ser considerada uma violação de restrições. Caso se materialize uma condição que não permita valores não nulos, será a restrição *NOT NULL*. Os valores só são considerados omissos, caso as entidades que deveriam ter valores mensuráveis não os tenham.

- **Registros omissos:** Acontecem quando não há entidades completas e que não estão representadas por um registro no conjunto de dados.

Limpar dados pode ser um processo demorado, dependendo de como os dados foram registrados. É preciso seguir os passos de inspecionar, limpar e verificar para que as anomalias no banco de dados sejam devidamente corrigidas. Durante a limpeza de dados são deletados os dados que não se quer analisar, os dados irrelevantes e os dados duplicados. É interessante limpar primeiramente os dados duplicados porque é recorrente que registrem os mesmos dados mais de uma vez na hora da coleta. Já os dados que não se deseja, precisam ser deletados do banco de dados com bastante atenção, pois

eles muitas vezes estão corretos, mas não estão de acordo com o problema específico analisado. Isso não quer dizer que não se pode utilizar esses dados futuramente para tentar analisar outro problema, então é muito importante manter uma cópia salva com todos os dados coletados.

2.7 Transformação de dados

A transformação de dados consiste na modificação da estrutura ou do formato dos dados para torná-los mais adequados para análise (HAND; MANNILA; SMYTH, 2001).

As técnicas de transformação de dados mais comuns incluem:

- **Normalização:** a normalização consiste em converter os dados para uma escala comum, como por exemplo, a escala de 0 a 1.
- **Escalamento:** o escalamento consiste em ajustar os dados para que fiquem dentro de um intervalo específico, como por exemplo, entre 0 e 100.
- **Discretização:** a discretização consiste em converter os dados contínuos em dados discretos.
- **Categorização:** a categorização consiste em atribuir rótulos aos dados

2.8 Redução de dimensionalidade

A redução de dimensionalidade consiste na redução do número de atributos dos dados para melhorar o desempenho dos algoritmos de mineração (HAND; MANNILA; SMYTH, 2001).

As técnicas de redução de dimensionalidade mais comuns incluem:

- **Redução de componentes principais:** a redução de componentes principais consiste na identificação de um conjunto de atributos que explicam a maior parte da variância dos dados.
- **Agrupamento:** o agrupamento consiste em agrupar os dados de acordo com suas semelhanças.
- **Redundância:** a redundância consiste na identificação de atributos que são redundantes.

A escolha das técnicas de pré-processamento a serem utilizadas depende do tipo de dados e do objetivo da análise.

2.9 Weka

O WEKA (*Waikato Environment for Knowledge Analysis*) é um *software* de código aberto emitido sob Licença Pública Geral (GNU). O *software* é um coleção de algoritmos de aprendizagem de máquina para tarefas de mineração de dados que contém ferramentas para preparação de dados, associação, classificação, regressão, agrupamento,mineração de regras de associação e visualização. O Weka está disponível para as plataformas Windows, Linux e MacOS (WAIKATO, 2023). Após a instalação, não é preciso fazer nenhuma configuração adicional para começar a usá-lo a não ser que se deseje utilizar algoritmos específicos e acesso a banco de dados através de JDBC.

2.9.1 Arquivo CSV

O WEKA faz a leitura de arquivos *CSV* que vem do inglês *"comma-separated-values"* (valores separados por vírgula). Isso quer dizer que os campos de dados indicados neste formato são separados ou delimitados por uma vírgula (BROWNLEE, 2020). Para entender melhor, deve-se dar uma olhada na Tabela 2.

Tabela 2 – Disposição dos dados em um arquivo *CSV*

Nome	Ano	Estado
Maria	2018	Amazonas
João	2019	São Paulo
Miguel	2021	Acre

É possível ler estes dados em um arquivo *CSV* separados por vírgulas e por um espaçamento de linha da seguinte forma:

Maria,2018,Amazonas

João,2019,São Paulo

Miguel,2021,Acre

Esse tipo de arquivo é utilizado para armazenar dados que podem ser importados e exportados por diversos aplicativos de edição como *Microsoft Excel*, *Google Sheets*, *AppleNumbers* entre outros.

2.9.2 Métodos implementados no WEKA

- **Métodos de classificação:** São métodos utilizados em conjuntos de dados que já estão classificados para construir modelos preditivos, prevendo a classificação para dados futuros.
- **Métodos de predição numérica:** São métodos que preveem variáveis numéricas com base no valor de outras também numéricas.
- **Métodos de agrupamento:** São métodos usados para encontrar estrutura de grupos nos dados. Estes grupos contem objetos que podem compartilhar propriedades

e características relevantes para o domínio dos dados estudados.

- **Métodos de associação:** São métodos que se baseiam em cálculos estatísticos de frequência em uma determinada base de dados. Dessa maneira é possível medir uma variável de confiança.

2.10 Métodos utilizados no WEKA

Nesta sessão é explicado como funcionam os percentuais de treino e validação padrão no WEKA. Também é explicado os métodos utilizados no WEKA para o desenvolvimento deste trabalho. Os métodos utilidados foram: *Apriori* para correlação de variáveis e em seguida os métodos *Decision Tree*, *Naive Bayes*, *One Rule*, *K-Nearest Neighbor*, *Random Subspace* e *Bagging* que são métodos classificadores.

2.10.1 Percentuais de treino e validação cruzada no WEKA

No WEKA, os percentuais de treino e validação cruzada são usados para dividir os dados em dois conjuntos, um para treinamento e outro para validação. O conjunto de treinamento é usado para treinar o modelo de aprendizado de máquina, enquanto o conjunto de validação é usado para avaliar o desempenho do modelo (WAIKATO, 2023). O percentual de treino é a porcentagem dos dados que é usada para treinamento. O percentual de validação é a porcentagem dos dados que é usada para validação. O percentual de validação padrão é **25%**, mas pode ser alterado. Para dividir os dados em conjuntos de treino e validação, o WEKA usa um método chamado *holdout*. O método holdout funciona da seguinte forma:

1. Os dados são embaralhados.
2. Uma porcentagem dos dados é selecionada para o conjunto de validação.
3. Os dados restantes são selecionados para o conjunto de treinamento.

Por exemplo, se o percentual de treino for 75%, 75% dos dados serão selecionados para o conjunto de treinamento e 25% dos dados serão selecionados para o conjunto de validação.

A validação cruzada é uma técnica que pode ser usada para melhorar a precisão da avaliação do desempenho do modelo (WITTEN; FRANK; HALL, 2016). A validação cruzada funciona da seguinte forma:

1. Os dados são divididos em k conjuntos de validação.
2. O modelo é treinado em k-1 conjuntos de validação.
3. O desempenho do modelo é avaliado no conjunto de validação restante.

Por exemplo, se k for 10, os dados serão divididos em 10 conjuntos de validação. O modelo será treinado em 9 conjuntos de validação e avaliado no conjunto de validação restante.

Weka fornece várias opções de validação cruzada, incluindo:

- ***Cross-validation (CV):*** o método de validação cruzada padrão.
- ***Repeated CV:*** o método de validação cruzada repetida.
- ***Leave-one-out CV:*** o método de validação cruzada *leave-one-out*.

Para este estudo de caso, foi utilizado o método de validação cruzada padrão.

2.10.2 Método de correlação *Apriori*

O método de correlação *Apriori* é um algoritmo de Aprendizagem de Máquina que pode ser usado para identificar correlações entre variáveis. O algoritmo é baseado na ideia de que as variáveis que são correlacionadas tendem a ocorrer juntas. O método de correlação *Apriori* funciona da seguinte forma:

1. O algoritmo começa com um conjunto de dados de treinamento.

2. O algoritmo calcula as correlações entre todas as pares de variáveis no conjunto de dados.

3. O algoritmo identifica os pares de variáveis com as correlações mais altas.

4. O algoritmo retorna os pares de variáveis identificados.

Este algoritmo é simples e fácil de implementar, também é muito rápido, eficiente, robusto a ruídos e *outliers* (valores que se desviam significativamente dos outros valores de uma distribuição).

2.10.3 *Decision Tree*

As Árvore de Decisões ou *Decision Tree* são algoritmos de *Machine Learning* largamente utilizados (CRISTIANINI; SHAWE-TAYLOR, 2020). Com uma estrutura simples e eficaz, geralmente apresentam bons resultados para previsões. A função da árvore de decisão particiona recursivamente um conjunto de treinamentos até que cada subconjunto obtido desse particionamento contenha casos de uma única classe tomando como entrada um objeto ou situação descrito por um conjunto de atributos e retorna uma decisão que é um valor de saída previsto de acordo com a entrada. Tais atributos podem ser discretos ou contínuos. Na construção da árvore é aplicado o critério de divisão e conquista e também o critério guloso para que sejam escolhidas as melhores partições e os melhores atributos resultando numa árvore de classificação que é a resposta para uma sequência ordenada de perguntas. As perguntas feitas a cada passo dessa sequência dependem das respostas das perguntas anteriores. Quanto a estrutura da árvore de decisão, tem-se o ponto de partida que é chamado de nó raiz e fica localizado no topo da árvore. Sendo um nó um subconjunto de atributos, ele pode ser um nó terminal ou nó não-terminal(HASTIE et al., 2009). Quando o nó é não-terminal, o nó se divide entre nós-filhos. Essa divisão é determinada de acordo com a condição que está sobre o valor de um único atributo e dividirá os exemplos de

acordo com a condição estabelecida em outros nós. Quando o nó não se divide ele é um nó terminal e para ele é atribuído a uma classe. Na Figura 2, está o exemplo de árvore de decisão.

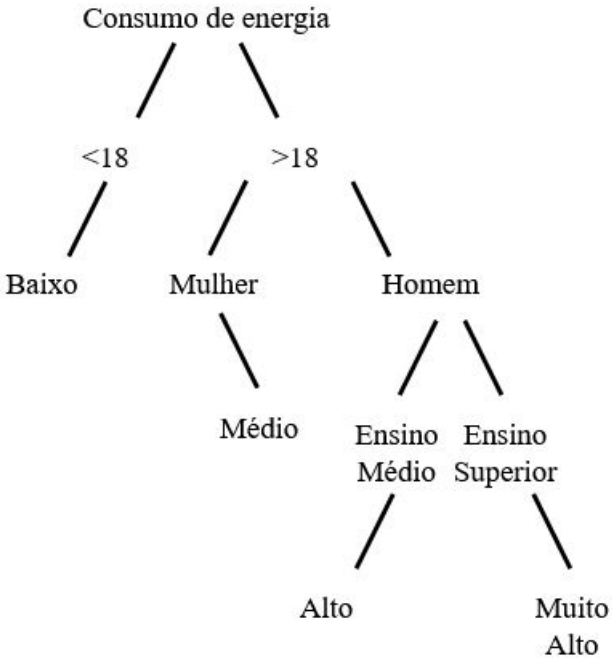

Figura 2 – Exemplo de árvore de decisão

Esta árvore de decisão é usada para prever o consumo de energia de uma pessoa com base em sua idade, gênero e escolaridade. A árvore começa com o atributo idade. Se a pessoa tiver menos de 18 anos, ela é classificada como "Baixo". Se a pessoa tiver mais de 18 anos, a árvore continua para o atributo gênero. Se a pessoa for do sexo feminino, ela é classificada como "Médio". Se a pessoa for do sexo masculino, a árvore continua para o atributo escolaridade. Se a pessoa tiver ensino médio, ela é classificada como "Alto". Se a pessoa tiver ensino superior, ela é classificada como "Muito alto".

Aqui está uma explicação de como a árvore de decisão funciona:

- O nó raiz da árvore representa a classe "Consumo de energia".
- O nó filho esquerdo representa o caso de a pessoa ser do sexo feminino.
- O nó filho direito representa o caso de a pessoa ser do sexo masculino.

- O nó filho esquerdo do nó filho esquerdo representa o caso de a pessoa ter menos de 18 anos.
- O nó filho direito do nó filho esquerdo representa o caso de a pessoa ter mais de 18 anos.
- O nó filho esquerdo do nó filho direito representa o caso de a pessoa ter ensino médio.
- O nó filho direito do nó filho direito representa o caso de a pessoa ter ensino superior.

Os algoritmos de árvore de decisão implementado no WEKA são os algoritmos J48 e C4.5 que são algoritmos de aprendizado de máquina de indução de árvores de decisão. Eles são baseados no algoritmo ID3, desenvolvido por Ross Quinlan em 1986 (TZIRAKIS; TJORTJIS, 2016). O algoritmo J48 é uma implementação do algoritmo C4.5. Ele é um algoritmo de indução de árvores de decisão de propósito geral, que pode ser usado para uma variedade de problemas de classificação (QUINLAN, 1993). Os algoritmos J48 e C4.5 funcionam da seguinte forma:

1. Começam com um conjunto de dados de treinamento.
2. Criam uma árvore de decisão vazia.
3. Para cada nó da árvore:

   a. Selecionam a característica que fornece o maior ganho de informação.
   b. Dividem o nó em subnós, um para cada valor da característica selecionada.
   c. Repetem os passos 3 e 4 para cada subnó.
4. O processo termina quando todos os nós da árvore são folhas, ou seja, quando não há mais subnós a serem criados.

2.10.4 *Naive Bayes*

O classificador *Naive Bayes* é usado para prever a probabilidade de determinados dados pertencerem a uma classe em particular. Para fazer tal previsão é usando o Teorema de Bayes. Então, considerando dois conjuntos de eventos aleatórios A e B, a probabilidade de se determinar A a partir do que se sabe de B pode ser obtida através da Equação (2.1).

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)} \tag{2.1}$$

Onde:

$P(A_i, ...A_n)$ equivalem a classes diferentes.
$P(A_i|B)$ significa a probabilidade de B acontecer já que A se confirmou.
$P(A_i)$ é a probabilidade de A acontecer.
$P(B)$ é a probabilidade de B acontecer.

Por ser simples e rápido, o desempenho desse algoritmo é relativamente maior do que outros classificadores. Então o algoritmo *Naive Bayes* precisa de uma amostra de dados pequena para concluir um teste de classificação com uma boa precisão.

2.10.5 *One Rule*

O *OR* ou *OneR* é um algoritmo de classificação simples e objetivo que gera uma regra para cada predição no banco de dados. Então, o algoritmo seleciona a regra que tem a menor quantidade de erros e por isso "*One Rule*". Para criar uma regra de predição, é construída uma tabela de frequência para cada preditor o alvo. Para cada preditor, em cada valor da predição faz uma regra passando pelos seguintes passos:

1. Conta a frequência que cada valor alvo (classe) aparece.
2. Encontra a classe mais frequente.
3. Faz a regra atribuir a essa classe o valor do preditor.

4. Calcula o total de erros para as regras em cada preditor.

5. Escolhe o preditor com o menor valor de erros.

2.10.6 *K-Nearest Neighbor*

O K-Vizinhos mais próximos ou *K-Nearest Neighbor* (*KNN*) é uma algoritmo de aprendizagem baseado em instâncias que pertence ao grupo de algoritmos *Instance-Based Learning* (*IBL*). Os algoritmos desse grupo tem a função de armazenar todas as instâncias de treinamento e quando surge uma nova instância para ser classificada, um conjunto conjunto de instâncias similares à essa nova instância é recuperada do conjunto de treinamento utilizada para classificá-la (FARIA, 2016). O *KNN* é um método de aprendizagem supervisionada, do tipo classificador, não-paramétrico, que utiliza *Lazy Learning* e possui três elementos principais: um conjunto de exemplos rotulados (por exemplo, um conjunto de registros armazenados), uma métrica de distância, e o valor de k (o número de vizinhos mais próximos) (OLIVEIRA, 2016). Para ilustrar o funcionamento do algoritmo *KNN*, tem-se a Figura 3 onde nota-se uma instância que será classificada e é representada pela interrogação. As instâncias de treinamento já estão classificadas e associadas as classes triângulo e a classe quadrado.

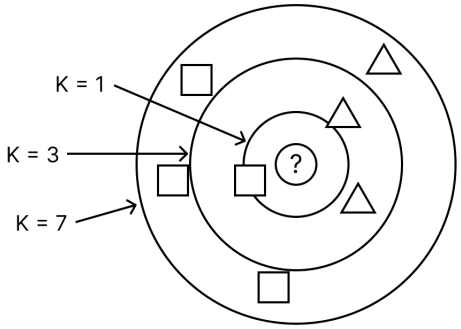

Figura 3 – Funcionamento do *KNN*

No exemplo da Figura 3, pela maneira que o algoritmo *KNN* funciona, para K=1, a instância recebe a classificação como pertencente a classe quadrado, pois é o vizinho

mais próximo. Para K=3 a classificação será como pertencente a classe triângulo, pois duas instâncias dos três vizinhos que estão mais próximos são pertencentes a classe triângulo e apenas um tem a classe quadrado. Por fim, para k=7, a nova instância será a classe quadrado.

2.10.7 *Random Subspace*

O *Random Subspace* (*RS*) ou subespaços aleatórios é um método em que os classificadores são construídos em um subconjunto de características dos dados disponíveis de tamanho pré-definido, amostras aleatoriamente(KUNCHEVA, 2014). Para construir um conjunto R, são coletadas L amostras de um tamanho M, extraídas de uma distribuição uniforme X, sem substituição, ou seja, o objeto selecionado para a mostra é removido do conjunto de dados original. Cada conjunto de características representa um subespaço da distribuição X de tamanho M. O algoritmo random subspace funciona da seguinte forma:

1. Começa com um conjunto de dados de treinamento.
2. Para cada árvore, seleciona aleatoriamente um subconjunto de características do conjunto de dados.
3. Constrói uma árvore de decisão usando o subconjunto de características selecionado.
4. Repete os passos 2 e 3 para construir um número especificado de árvores.
5. Para classificar um novo exemplo, volta em cada árvore para determinar a classe do exemplo.

2.10.8 *Bagging*

O algoritmo *Bagging*, ou *Bootstrap Aggregating*, é um método de Aprendizagem de Máquina Supervisionada *ensemble* que combina vários modelos de Aprendizagem de Máquina simples para melhorar a precisão e a generalização (BREIMAN, 1996). Para fazer isso, o *Bagging* amostra repetidamente o conjunto de dados de treinamento com reposição, o que significa que alguns exemplos podem ser selecionados mais de uma vez. Em cada amostra, um modelo de Aprendizagem de Máquina é construído. Os resultados dos modelos são então combinados para produzir uma previsão final. O algoritmo bagging funciona da seguinte forma:

1. Começa com um conjunto de dados de treinamento.
2. Para cada modelo, seleciona aleatoriamente um subconjunto de dados do conjunto de dados de treinamento com reposição.
3. Constrói um modelo usando o subconjunto de dados selecionado.
4. Repete os passos 2 e 3 para construir um número especificado de modelos.
5. Para classificar um novo exemplo, volta em cada modelo para determinar a classe do exemplo.

2.11 *Google Colaboratory*

O *Google Colaboratory* é uma plataforma de computação em nuvem que permite aos usuários trabalhar com linguagem de programação *Python* e seus pacotes diretamente no navegador, incluindo os utilizados para Aprendizagem de Máquina (GOOGLE, 2023). Na Figura 4 vê-se a interface da plataforma. Nela pode-se ver a célula de código usada para escrever e executar códigos em *Python*.

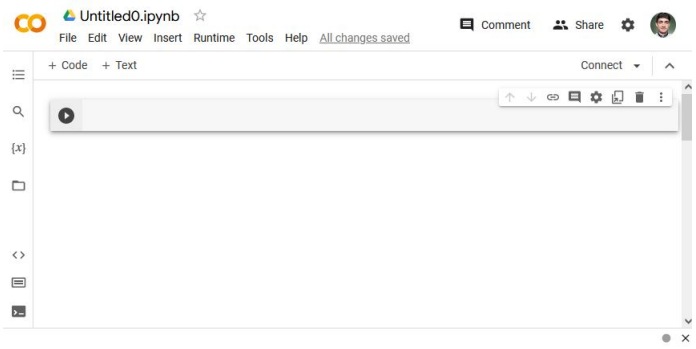

Figura 4 – Interface do *Google Colaboratory*

O *Google Colaboratory* é uma ferramenta poderosa para Aprendizagem de Máquina Supervisionada pois oferece uma variedade de recursos que facilitam o desenvolvimento e a implantação de modelos de Aprendizagem de Máquina. Os recursos incluem:

- **Classificação:** usado para treinar modelos de classificação para classificar dados em categorias.
- **Regressão:** usado para treinar modelos de regressão para prever valores numéricos.
- **Reconhecimento de padrões:** usado para treinar modelos de reconhecimento de padrões para identificar padrões em dados.

A ferramenta é poderosa, gratuita e acessível, oferecendo uma variedade de recursos que facilitam o desenvolvimento e a aplicação de modelos de Aprendizagem de Máquina Supervisionada.

2.12 Linguagem *Python*

O *Python* é uma linguagem de programação de alto nível, de uso geral, interpretada, de código aberto, e com sintaxe de fácil aprendizagem. É uma linguagem versátil

que pode ser usada para uma ampla gama de tarefas, incluindo Ciência de Dados, Análise de Dados, e Aprendizagem de Máquina(IBM, 2019).

2.12.1 *PANDAS*

A biblioteca *PANDAS* é uma ferramenta de código aberto para análise de dados em *Python*. Ela fornece uma abordagem rápida e flexível para trabalhar com dados relacionais (ou rotulados), de maneira simples e intuitiva (KINNEY, 2017).

O nome *PANDAS* é derivado do termo *Panel Data*, um conceito em inglês relacionado ao campo de estudo da econometria. Apesar de ser associado ao mamífero da família de ursos, assim como o *Python* é associado com a espécie de cobra erroneamente, o nome da biblioteca não tem relação com esses animais.

O *PANDAS* também possui ótima integração com várias outras bibliotecas muito utilizadas em Ciência de Dados, como *Numpy, Scikit-Learn, Seaborn, Altair, Matplotlib, Plotly, Scipy* e outros.

2.12.2 *Scikit-learn*

O Scikit-learn é uma biblioteca de código aberto para aprendizado de máquina em Python. Ele fornece uma ampla gama de algoritmos de aprendizado de máquina, incluindo classificação, regressão, agrupamento, redução de dimensionalidade e aprendizagem de reforço (PEDREGOSA, 2011).

O Scikit-learn é uma ferramenta poderosa para cientistas de dados e engenheiros de *machine learning*. É fácil de aprender e usar, e oferece uma ampla gama de recursos para trabalhar com dados de diferentes tipos e tamanhos.

2.12.3 Percentuais de treino e validação cruzada no *Scikit-learn*

Os valores padrão do método *train_test_split* do *Scikit-learn* para os percentuais de treino e validação cruzada são os seguintes:

- Percentual de treino: 75%
- Percentual de validação: 25%

Isso significa que, por padrão, o método **train_test_split** dividirá os dados em dois conjuntos, um com 75% dos dados para treinamento e outro com 25% dos dados para validação.

O percentual de treino pode ser ajustado usando o argumento *test_size*. Por exemplo, para definir o percentual de treino para 80%,pode-se usar o seguinte código da Figura 5:

```
from sklearn.model_selection import train_test_split

X, y = load_data()

X_train, X_test, y_train, y_test = train_test_split( X, y,
test_size=0.2)
```

Figura 5 – Definição de percentual de treino

O percentual de validação pode ser ajustado usando o argumento *train_size*. Por exemplo, para definir o percentual de validação para 10%, pode-se usar o seguinte código da Figura 6:

```
from sklearn.model_selection import train_test_split

X, y = load_data()

X_train, X_test, y_train, y_test = train_test_split(X, y,train_size=0.9)
```

Figura 6 – Definição de percentual de validação

É importante ressaltar que os valores padrão do *train_test_split* podem não ser adequados para todos os casos. Por exemplo, se o conjunto de dados for pequeno, pode

ser melhor usar um percentual de treino menor para evitar *overfitting*. Por outro lado, se o conjunto de dados for grande, pode ser melhor usar um percentual de treino maior para obter uma estimativa mais precisa do desempenho do modelo.

O valor padrão de percentual de treino para validação cruzada no *Scikit-learn* é 75%. Isso significa que, por padrão, a validação cruzada dividirá os dados em k-1 conjuntos de treino e 1 conjunto de validação, sendo k o número de *folds*.

Por exemplo, se k=10, então os dados serão divididos em 9 conjuntos de treino e 1 conjunto de validação. O modelo será treinado em 9 conjuntos de treino e avaliado no conjunto de validação restante.

O percentual de treino pode ser ajustado usando o argumento **train_size** da função *cross_val_score()*. Por exemplo, para definir o percentual de treino para 80%, pode-se usar o seguinte código da Figura 7:

```
from sklearn.model_selection import cross_val_score

X, y = load_data()

clf = LogisticRegression()

scores = cross_val_score(clf, X, y, cv=10, train_size=0.8)
```

Figura 7 – Definição de percentual de treino e validação cruzada

É importante ressaltar que o valor padrão de percentual de treino para validação cruzada pode não ser adequado para todos os casos. Por exemplo, se o conjunto de dados for pequeno, pode ser melhor usar um percentual de treino menor para evitar *overfitting*. Por outro lado, se o conjunto de dados for grande, pode ser melhor usar um percentual de treino maior para obter uma estimativa mais precisa do desempenho do modelo.

2.13 Acurácia

Os valores de acurácia podem variar de acordo com uma série de fatores. No entanto, é possível obter um valor de acurácia elevado, desde que os dados de treinamento

sejam de alta qualidade e o modelo seja bem ajustado (MURPHY, 2012).

Alguns fatores que podem afetar o valor de acurácia são:

- **Número de features:** Quanto maior o número de *features*, mais complexo o modelo e, consequentemente, menor o valor de acurácia esperado.
- **Interdependência das *features***: Se as *features* estiverem inter-relacionadas, o modelo pode ser mais preciso.
- **Distribuição dos dados:** Se os dados estiverem distribuídos de forma homogênea, o modelo pode ser mais preciso.

Ao trabalhar com problemas complexos, é importante considerar esses fatores para melhorar o desempenho do modelo.

Exemplos de valores de acurácia para problemas complexos:

- **Reconhecimento de imagens:** 70% ou mais
- **Detecção de fraudes:** 80% ou mais
- **Classificação de texto:** 60% ou mais
- **Diagnóstico médico:** 75% ou mais

Para o estudo de caso considera-se valores de acurácia com classificação de 60% ou mais pois a base de dados é pequena, mas o estudo é complexo e testa a correlação de muitos atributos simultaneamente. Neste estudo, somente a acurácia dos classificadores foi utilizada como parâmetro de comparação entre os algoritmos.

3

MEDODOLOGIA

Neste capítulo detalha-se como o embasamento teórico foi aplicado no trabalho proposto. Antes de iniciar a Análise dos Dados, foi importante fazer o pré-processamento dos dados para remover valores nulos, *outliers* e outros problemas. Também foi feito o enriquecimento dos dados através da criação de duas novas variáveis: potência instalada e consumo de energia diário. Para avaliar a correlação de potência instalada e consumo diário de energia, foi utilizado o algoritmo *Apriori* no WEKA. Para avaliar o desempenho de diferentes métodos de previsão, foram feitas diferentes abordagens utilizando diferentes algoritmos no WEKA. Depois da obtenção dos resultados no WEKA, os mesmos métodos foram executados Linguagem de Programação em *Python*, no *Google Colaboratory* com as bibliotecas *PANDAS* e *Scikit-learn* para comparação de desempenho das duas ferramentas. Já se pensando em escalabilidade da base de dados, a escolha e preferência pela Linguagem *Python* se justifica.

3.1 Dados utilizados

Os banco de dados utilizado foi coletado de uma subárea da área denominada Mosaico do Baixo Rio Negro, que engloba 11 unidades de conservação, localizadas entre os municípios de Manaus, Novo Airão, Iranduba, Barcelos e Manacapuru. Especificamente, os questionários foram aplicados e coletados em 14 comunidades não

eletrificadas na Área de Proteção Ambiental (APA) do Rio Negro, da Reserva de Desenvolvimento Sustentável (RDS) do Rio Negro e da Reserva de Desenvolvimento Sustentável (RDS) da Puranga Conquista. Os questionários cobriram informações individuais por família, relacionadas a questões de moradia, nível de escolaridade, estrutura familiar, atividades produtivas, saúde, necessidades e aspirações, uso de energéticos diversos, demanda elétrica e renda familiar, totalizando 56 macro-questões e 179 respondentes.

3.1.1 Pré-processamento dos dados

O banco de dados apresentava todos os tipos de anomalias possíveis e primeiramente foi preciso limpá-lo com bastante atenção. A redução de dimensionamento de componentes principais foi feita e agrupamento dos dados semelhantes também, juntamente com a identificação e redução e remoção dos atributos redundantes. Também foi feita a transformação dos dados através da categorização, dessa maneira, os atributos foram preenchidos com dados categóricos. Essa foi a parte mais trabalhosa de trabalhar com dados e também a parte que demandou mais tempo. Durante o pré-processamento dos dados, enriquece-se a base de dados gerando, a partir dos dados já coletados, outros dados relevantes. Esse é o caso da criação das variáveis potência instalada e consumo de energia diário (estimado).

Para a Potência Instalada fez-se o cálculo da potência nominal para todos os aparelhos eletrônicos registrados em cada instância, como lampadas, TVs, rádios entre outros. A potência instalada foi obtida utilizando planilha no *Excel*. O cálculo pode ser representado pela seguinte fórmula:

$$P_{Instalada} = (P_i + ... + P_n) \tag{3.1}$$

Onde:

$P_{Instalada}$ é a potencia total numa instância.

$(P_i + ... + P_n)$ é a soma de todas as potências nominais presentes na instância.

Para o Consumo de Energia Diário, foi feito o levantamento da estimativa da curva de carga diário utilizando a planilha do *Excel* para estimar as horas do dia em que os aparelhos eletrodomésticos e lampadas foram utilizados. Com essas informações foi possível calcular a quantidade de energia consumida para cada instância.

Tabela 3 – Potencia instalada e estimativa de consumo de energia diário

Aparelhos	Quant.	Potência (W)	Uso (h)	Consumo (KWh)
Aparelho de som	1	110	6	0,66
Freezer	1	66	11	1,58
Lampadas	4	23	11	1,10
Lava Roupas	1	147	1	0,14
Liquidificador	1	213	0,6	0,10
Carregador de celualar	1	10	1	0,02
TV 42 (LED)	1	203	11	2,03
Ventilador de mesa	1	72	11	0,93
Satelite TV	1	75	11	0,75
Ferro de Passar	1	1050	11	1,05

Para realizar o cálculo do Consumo (KWh), foi utilizada a seguinte expressão:

$$P_{Consumo} = \frac{P_{nominal} \cdot N_{Horas}}{1000} \tag{3.2}$$

Onde:

$P_{Consumo}$ equivale ao consumo diário.

$P_{Nominal}$ equivale a potencia nominal do aparelho.

N_{Horas} equivale ao tempo de horas de uso.

Para esta tabela o consumo total diário é de 8,36 kWh e a potência é de 1969 W.

3.2 Correlação de dados no WEKA

O WEKA foi utilizado para os testes iniciais. Com o *software*, foi obtida a correlação entre as variáveis que pode ser vista na Figura 8. A correlação foi feita com o método *Apriori* que é um algoritmo de aprendizagem de máquina que pode ser usado para identificar correlações entre variáveis.

O teste de correlação retornou um valor de 0.9 de confiabilidade, o que é um valor alto, mas deve-se levar em consideração que se está trabalhando com uma base de dados pequena.

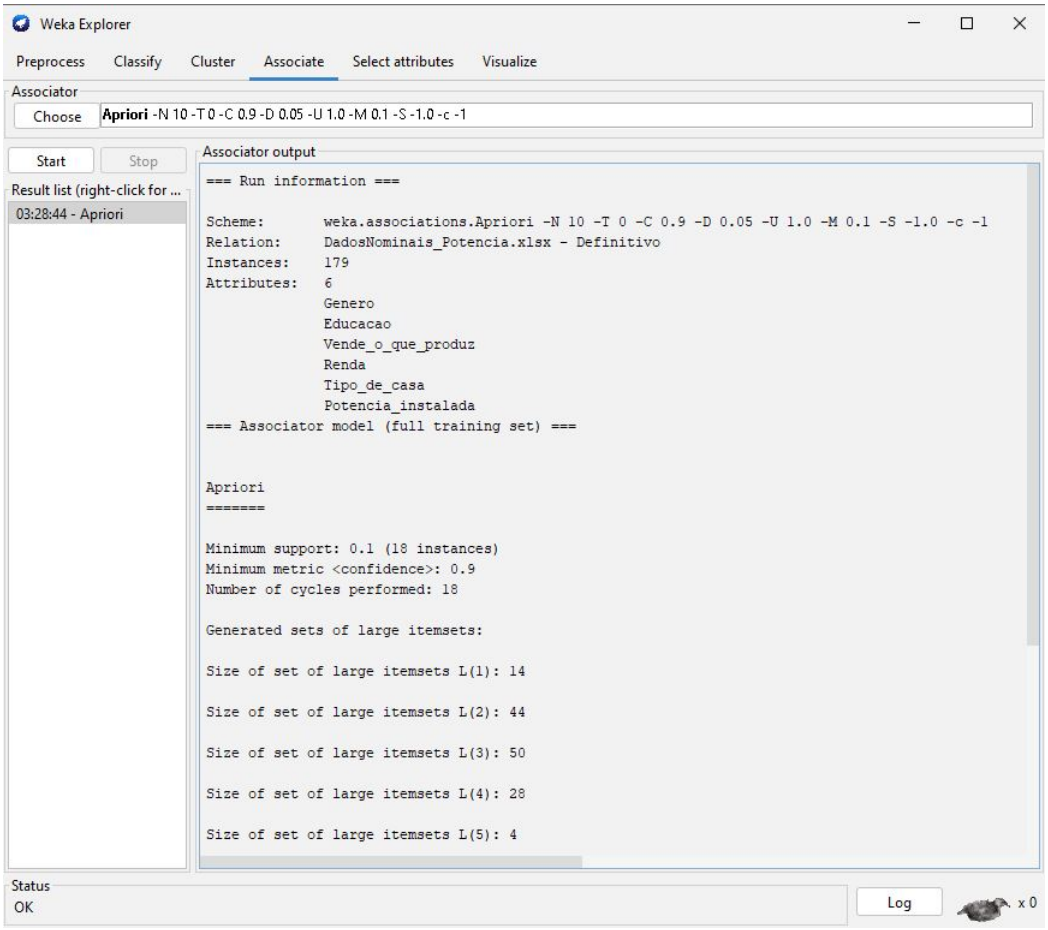

Figura 8 – Método de Correlação Apriori

Desta maneira, foi possível selecionar as variáveis correlacionadas que foram: Gênero, Educação, Vende a produção, Renda, Tipo de casa, Potência instalada e Consumo de energia diário. Primeiramente, foram utilizados os seguintes algoritmos: *Decision Tree, Naive Bayes, One Rule e KNN*. Entretanto, outros dois algoritmos foram adicionados pois apresentaram resultados satisfatórios. Esses algoritmos foram: *Random Subspace e Bagging*.

3.3 Teste de métodos no WEKA

Na fase de testes utilizou-se todos os métodos classificadores disponíveis e funcionais para uma base de dados nominal, filtrando os classificadores mais coerentes para o uso de Aprendizagem de Máquina Supervisionada, procurando resultados satisfatórios. Na Figura 9, é possível ver todos os métodos disponíveis para testes de classificação. Alguns métodos foram desabilitados pelo programa pois usou-se uma base de dados nominal.

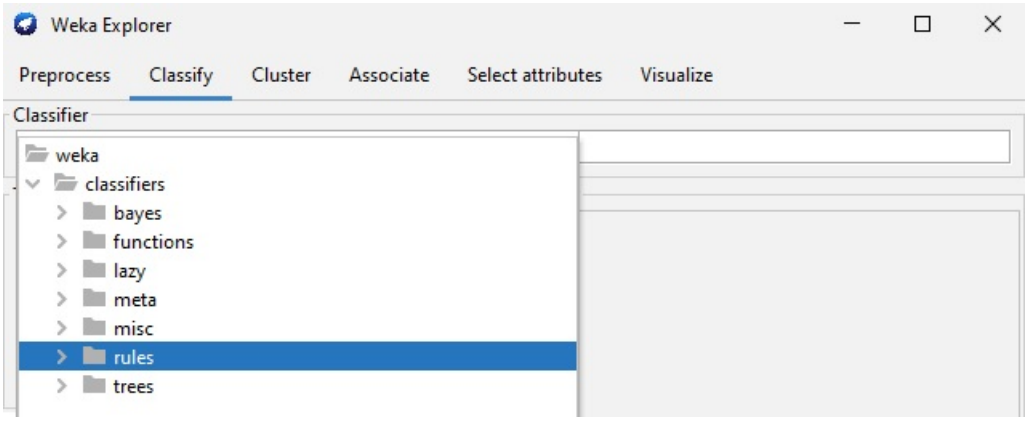

Figura 9 – Teste de métodos classificadores

3.3.1 Testes de previsão para potência e energia no WEKA

Os teste foram feitos para prever a variáveis Potência instalada e Consumo de energia diário. Os parâmetros usados para o *Percent split* foram 30%, 50% e 70%, como percentual de treino e 25% como percentual de validação que é o valor *default*. Para o *Cross Validation*, foram utilizados *n-folds* de 3, 5 e 7, para validação e 30%, 50% e 70% como percentual de treino.

3.3.1.1 Classificador *Decision Tree*

O classificador *Decision Tree* utilizado no WEKA foi o J48, uma árvore de decisão que é um aprimoramento do algoritmo C4.5. Este algoritmo funciona construindo uma árvore de decisão, na qual cada nó representa uma decisão e cada ramo representa uma possível saída.

Percent split (%)	Acurácia (%)	Cross Validation (n-folds)	Acurácia (%)
30	68,8	3	65,4
50	68,5	5	65,4
70	66,7	7	65,4

Tabela 4 – Potencia intalada com *Decision Tree*

Para a potencia instalada, nota-se na Tabela 4 que o *Percent Split* se destacou com os melhores resultados. Com 30% da amostra treinada, obteve-se uma acurácia de 68,8%. Com 50% da amostra treinada, obteve-se 68,5% de acurácia.

Percent split (%)	Acurácia (%)	Cross Validation (n-folds)	Acurácia (%)
30	67,2	3	42,2
50	68,5	5	41,6
70	64,8	7	44,4

Tabela 5 – Consumo de energia diário com *Decision Tree*

Para o consumo de energia diário, ve-se na Tabela 5 que o *Percent Split* se destacou com os melhores resultados. Com o treinamento de 30% e 50% do banco de dados obtém-se acurácia de 67,2% e 68,5% respectivamente.

3.3.1.2 Classificador *Naive Bayes*

O algoritmo de classificação *Naive Bayes* é baseado na suposição de que características são independentes entre si.

Percent split (%)	Acurácia (%)	Cross Validation (n-folds)	Acurácia (%)
30	64,0	3	62,6
50	67,4	5	62,0
70	68,5	7	61,5

Tabela 6 – Potência instalada com *Naive Bayes*

Para potência instalada nota-se na Tabela 6 que o *Percent Split* destacou-se com os melhores resultados. Com 50% da amostra treinada, obteve-se 67,4% de acurácia.

Percent split (%)	Acurácia (%)	Cross Validation (n-folds)	Acurácia (%)
30	35,2	3	40,8
50	39,3	5	43,0
70	46,3	7	43,6

Tabela 7 – Consumo de energia diário com *Naive Bayes*

Para a o consumo de energia diário nota-se na Tabela 7 que o *Cross Validation* se destacou com os melhores resultados. Com 50% da amostra treinada, obteve-se 43,0% de acurácia.

3.3.1.3 *One Rule*

O algoritmo de classificação *OneRule* funciona de uma maneira bem simples, encontrando a regra com o menor erro de classificação.

Percent split (%)	Acurácia (%)	Cross Validation (n-folds)	Acurácia (%)
30	68,8	3	65,4
50	68,5	5	65,4
70	66,7	7	65,4

Tabela 8 – Potencia intalada para *One Rule*

Para a potência instalada nota-se na Tabela 8 que o *Percent Split* se destacou com os melhores resultados. Com 30% da amostra treinada, obteve-se 68,8% de acurácia. Para *n-folds*, todos os testes retornaram 65,4% de acurácia.

Percent split (%)	Acurácia (%)	Cross Validation (n-folds)	Acurácia (%)
30	36,0	3	44,1
50	38,2	5	44,7
70	42,6	7	44,7

Tabela 9 – Consumo de energia diário - One Rule

Para o consumo de energia diário, conforme Tabela 9, o *Cross Validation* se destacou com os melhores resultados. Com os valores *n-fold* = 3, 5 e 7 a acurácia foi de 44,1%, 44,7% e 44,7%, respectivamente.

3.3.1.4 *K-Nearest Neighbor*

O algoritmo *KNN* pode ser usado para classificação e regressão. Através dos k-vizinhos encontrados mais próximos de um ponto de dados ele pode prever ou classificar ponto de dados.

Percent split (%)	Acurácia (%)	Cross Validation (n-folds)	Acurácia (%)
30	60,0	3	58,7
50	57,3	5	58,1
70	61,1	7	57,0

Tabela 10 – Potencia intalada com *KNN*

Para a potência instalada vê-se na Tabela 10 que o *Percent Split*, de modo geral se destacou com os melhores resultados. Entretanto, o *n-fold* = 5 se saiu um pouco melhor que o *Percent Split* de 50%.

Percent split (%)	Acurácia (%)	Cross Validation (n-folds)	Acurácia (%)
30	36,7	3	34,6
50	34,8	5	38,5
70	38,9	7	35,8

Tabela 11 – Consumo de energia diário com *KNN*

Para o consumo de energia diário, nota-se na Tabela 11 que o *Percent Split* e o *Cross Validation* não mostraram uma diferença de valores discrepantes. Os valores permaneceram na faixa entre 34,6% e 38,9%.

3.3.1.5 *Random Subspace*

O *Random Subspace* é um algoritmo *enable* que combina várias previsões de modelos de *Machine Learning*, construindo vários modelos de aprendizado de máquina em subconjuntos aleatórios das características do conjunto de dados de treinamento combinando as previsões desses modelos.

Percent split (%)	Acurácia (%)	Cross Validation (n-folds)	Acurácia (%)
30	68,8	3	64,8
50	68,5	5	62,6
70	66,7	7	62,6

Tabela 12 – Potencia intalada com *Random Subspace*

Para a potência instalada vê-se na Tabela 12 que o *Percent Split* se destacou com os melhores resultados. Com o treinamento de 50% da amostra, obteve-se acurácia de 68,5%.

Percent split (%)	Acurácia (%)	Cross Validation (n-folds)	Acurácia (%)
30	42,4	3	45,3
50	48,3	5	45,8
70	44,4	7	45,3

Tabela 13 – Consumo de energia diário com *Random Subspace*

Para o consumo de energia diário nota-se, na Tabela 13, que o *Percent Split* teve o melhor resultado. Com o treinamento de 50% dos dados ele retornou uma acurácia de 48,3%. O *Cross Validation* não retornou valores discrepantes, com valores na faixa entre 45,3% e 45,8%.

3.3.1.6 *Bagging*

O *Bagging* é um método que pode ser usado para melhorar o desempenho de modelos de aprendizagem de máquina.

Percent split (%)	Acurácia (%)	Cross Validation (n-folds)	Acurácia (%)
30	67,2	3	64,8
50	68,5	5	62,6
70	64,8	7	62,6

Tabela 14 – Potencia instalada para Bagging

Para a potência instalada vê-se, na Tabela 14, que o *Percent Split* teve o melhor resultado. Com o treinamento de 50% da base de dados ele retornou uma acurácia de 68,5%. A melhor acurácia para o *Cross Validation* foi para o *n-fold* = 5 e 7, ambos retornaram 43,3% de acurácia.

Percent split (%)	Acurácia (%)	Cross Validation (n-folds)	Acurácia (%)
30	44,8	3	38,5
50	38,2	5	43,3
70	53,7	7	43,3

Tabela 15 – Consumo de energia diário para *Bagging*

Para o consumo de energia diário nota-se, na Tabela 15, que o *Percent split* teve os melhores resultados. Para 70% de treinamento no banco de dados ele retornou uma acurácia de 53,7%. A melhor acurácia para o *Cross Validation* foi para o *n-fold* = 5 e 7, ambos retornaram 43,3% de acurácia.

3.4 Testes no Google Colaboratory

No *Google Colaboratory* foram executados os mesmos métodos executados no WEKA. Entretanto, algumas modificações precisaram ser feitas no arquivo para que os testes pudessem ter sucesso. Foi feita a transformação dos dados nominais categóricos para dados numerais ordinais. Também foram tratados os valores não respondidos que foram preenchidos através da estratégia 'do mais frequente'. Por fim, com a base de dados padronizada, foram feitos os testes na Linguagem de Programação *Python*.

3.4.1 Transformação de dados

Para trabalhar com banco de dados na Biblioteca *PANDAS* com a finalidade de alcançar melhores resultados, foi preciso transformar os dados nominais categóricos por números ordinais. Primeiramente, verificou-se as informações do banco de dados, como pode ser observado na Figura 10.

```
DadosComunidades.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 179 entries, 0 to 178
Data columns (total 6 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Genero              177 non-null    object
 1   Educacao            176 non-null    object
 2   Vende_o_que_produz  157 non-null    object
 3   Renda               163 non-null    object
 4   Tipo_de_casa        178 non-null    object
 5   Potencia_instalada  174 non-null    object
dtypes: object(6)
memory usage: 8.5+ KB
```

Figura 10 – Informações gerais do banco de dados

Na analise dos dados, observou-se que os dados são do tipo objeto. Para solucionar este problema, usou-se codificador, como mostrado na Figura 11. O codificador transformou os dados categóricos nominais em dados ordinais. Os dados deixaram de ser do tipo objeto e foram transformados em tipo ponto flutuante.

```
from sklearn.preprocessing import OrdinalEncoder

Codificador = OrdinalEncoder()
Codificador.fit(DadosComunidades)

DadosComunidadesCategorizados = Codificador.fit(DadosComunidades)
DadosComunidadesCategorizados = Codificador.transform(DadosComunidades)
```

Figura 11 – Mapeamento de dados por faixa para dados numéricos

Também resolveu-se o problema com os dados não respondidos. A estratégia utilizada foi preencher os dados faltosos com os valores mais frequentes para cada variável (Figura 12).

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy='most_frequent')
imputer.fit(DadosComunidadesCategorizados)
```

Figura 12 – Preenchendo os dados nulos com valores mais frequentes

Então, obteve-se o Banco de Dados Padrão (BDP), que é a base utilizada para teste em vários algoritmos conforme Figura 13. O BDP é o banco de dados resultante das manipulações feitas para melhorar a qualidade do banco de dados. Ele é um banco de dados com valores categóricos ordinais numéricos com os valores nulos preenchidos com os valores mais frequentes em cada atributo, potanto, é um banco de dados completo e configurado para ser usado com linguagem de programação Python.

```
BDP.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 179 entries, 0 to 178
Data columns (total 6 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Genero              179 non-null    float64
 1   Educacao            179 non-null    float64
 2   Vende_o_que_produz  179 non-null    float64
 3   Renda               179 non-null    float64
 4   Tipo_de_casa        179 non-null    float64
 5   Potencia_instalada  179 non-null    float64
dtypes: float64(6)
memory usage: 8.5 KB
```

Figura 13 – Mapeamento de dados por faixa para dados numéricos

Após o mapeamento, foi separado os *inputs* (as entradas) e os *target* (alvo para previsão) que foram potência instalada e consumo de energia diário. Na Figura 14, pode-se observar o exemplo para potência instalada.

```
inputs = BDP.drop('Potencia_instalada', axis = 'columns')
target = BDP.Potencia_instalada
```

Figura 14 – Mapeamento de dados por faixa para dados numéricos

3.4.2 Testes de previsão para potência e energia com *Python*

Após o mapeamento dos dados, foi possível utilizar diversos métodos da biblioteca *Scikit-learn*. Foram utilizados modelos de treinamento com linguagem de programação *Python*. As Figuras a seguir mostrarão a implementação dos métodos e as tabelas mostraram os resultados obtidos nos testes feitos com os valores de porcentagem da amostra para validação (*Percent split*) de 30%, 50% e 70% para o percentual de treino e 25% para o percentual de validação que é o valor *default* dos métodos. Para o *Cross Validation*, *n-folds* de 3, 5 ou 7, a porcentagem de amosta também foi de 30%, 50% e 70% para a validação e percentual de treino foi 25%, que é o valor padrão.

3.4.2.1 Classificador *Decision Tree*

Na Figura 15, nota-se a implementação do treinamento da amostra e do modelo para o classificador *Decision Tree*. Com o valor de *score* pode-se obter a acurácia.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =
train_test_split(inputs,target,test_size=0.5)

from sklearn import tree
model = tree.DecisionTreeClassifier()

model.fit(inputs, target)

model.score(X_train,y_train)
```

Figura 15 – Valor de previsão para o classificador *Decision Tree*

Na Figura 16, nota-se a implementação da validação cruzada. Aplicou-se o método classificador de árvore de decisão e o valor da validação cruzada. A função *scores.mean()*, retorna a média dos vetores da validação que também é utilizada para obter a acurácia.

```
from sklearn.model_selection import cross_val_score

clf = tree.DecisionTreeClassifier()
scores = cross_val_score(clf, inputs, target, cv=5)

scores.mean()
```

Figura 16 – Valor de previsão para o *Decision Tree* com validação cruzada

Na Tabela 16 observa-se o *score* do classificador para potência. O melhor *score* se encontra no *train split* de 30% e 70% com o valor de 0,792 que pode ser convertido para 79,2% de acurácia.

train split (%)	score (%)	cross validation	scores
30	0,792	3	0,563
50	0,775	5	0,630
70	0,792	7	0,636

Tabela 16 – Classificador *Decision Tree* para potência

Na Tabela 17 observa-se o *score* do classificador para energia. O melhor *score* se encontra no *train split* de 30%, 50% e 70%, todos com o mesmo valor de 0,614 que pode ser convertido para 61,4% de acurácia.

train split (%)	score	cross validation	scores (%)
30	0,614	3	0,413
50	0,614	5	0,435
70	0,614	7	0,413

Tabela 17 – Classificador *Decision Tree* para energia

3.4.2.2 Classificador *Naive Bayes*

Na Figura 17 observa-se a implementação do treinamento da amostra para o classificador *Naive Bayes*. Com o valor do *score* pode-se obter a acurácia.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =
train_test_split(inputs,target,test_size=0.5)

from sklearn.naive_bayes import GaussianNB
model = GaussianNB()

model.fit(inputs,target)

model.score(X_train,y_train)
```

Figura 17 – Valor de previsão para o *Naive Bayes*

Na Figura 18, observa-se a implementação da validação cruzada. Passou-se os parâmetros para o classificador *Naive Bayes* e o valor da validação cruzada. No final do código, obteve-se a média dos vetores da validação cruzada.

```
from sklearn.model_selection import cross_val_score

clf = GaussianNB()
scores = cross_val_score(clf,inputs, target, cv=5)

scores.mean()
```

Figura 18 – Valor de previsão para o *Naive Bayes* com validação cruzada

Na Tabela 18 encontra-se o *score* do classificador para potência. O melhor *score* se encontra no *train split* de 30% com o valor de 0,759 que pode ser interpretado como 75,9% de acurácia.

train split (%)	score (%)	cross validation	scores
30	0,759	3	0,591
50	0,688	5	0,614
70	0,674	7	0,608

Tabela 18 – Classificador *Naive Bayes* para potência

Na Tabela 19, observa-se o *score* do classificador para energia. O melhor *score* se encontra no *train split* de 30% com o valor de 0,259 que pode ser interpretado como 25,9% de acurácia.

train split (%)	score	cross validation	scores
30	0,259	3	0,153
50	0,100	5	0,156
70	0,100	7	0,156

Tabela 19 – Classificador *Naive Bayes* para energia

3.4.2.3 Classificador *One Rule*

Na Figura 19 vê-se a implementação do treinamento da amostra para o classificador *One Rule*. Com o valor do *score* pode-se chegar na interpretação da acurácia.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =
train_test_split(inputs,target,test_size=0.5)

import numpy as np
from sklearn.multiclass import OneVsRestClassifier
from sklearn.svm import SVC
model = OneVsRestClassifier(SVC()).fit(inputs, target)

model.fit(inputs,target)

model.score(X_train,y_train)
```

Figura 19 – Valor de previsão para com *One Rule*

Na Figura 20 vê-se a implementação da validação cruzada. Passou-se os parâmetros para o classificador *One Rule* e o valor da validação cruzada. No final do código, obteve-se a média dos vetores gerados da validação cruzada.

```
from sklearn.model_selection import cross_val_score

clf = OneVsRestClassifier(SVC()).fit(inputs, target)
scores = cross_val_score(clf, inputs, target, cv=5)

scores.mean()
```

Figura 20 – Valor de previsão para o *One Rule* com validação cruzada

Na Tabela 20, observa-se o *score* do classificador para a potência. O melhor *score* se encontra no *train split* de 50% com o valor de 0,696 que pode ser interpretado como 69,6% de acurácia.

train split (%)	score	cross validation	scores
30	0,672	3	0,681
50	0,696	5	0,681
70	0,679	7	0,681

Tabela 20 – Classificador *One Rule* para potência

Na Tabela 21 vê-se o *score* do classificador para energia. O melhor *score* se encontra no *train split* de 30% com o valor de 0,488 que pode ser interpretado como 48,8% de acurácia.

train split (%)	score	cross validation	scores
30	0,488	3	0,435
50	0,460	5	0,418
70	0,471	7	0,407

Tabela 21 – Classificador *One Rule* para energia

3.4.2.4 Classificador *KNN*

Na Figura 21, observa-se a implementação do treinamento da amostra para o classificador *KNN*. Com o valor do *score* pode-se chegar na interpretação da acurácia. Foi usado k-vizinhos = 10, que é o valor padrão do algoritmo.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =
train_test_split(inputs,target,test_size=0.5)

from sklearn.neighbors import KNeighborsClassifier
model = KNeighborsClassifier()

model.fit(inputs,target)

model.score(X_train,y_train)
```

Figura 21 – Valor de previsão para o KNN

A implementação da validação cruzada pode ser observada na Figura 22. Passou-se os parâmetros para o classificador *KNN* e o valor da validação cruzada. No final do código, obteve-se a média dos vetores gerados da validação cruzada.

```
from sklearn.model_selection import cross_val_score

clf = KNeighborsClassifier()
scores = cross_val_score(clf, inputs, target, cv=5)

scores.mean()
```

Figura 22 – Valor de previsão para o *KNN* com validação cruzada

Na Tabela 22 vê-se o *score* do classificador para a potência. O melhor *score* se encontra no *train split* de 30% com o valor de 0,740 que pode ser interpretado como 74,0% de acurácia.

train split (%)	score	cross validation	scores
30	0,740	3	0,642
50	0,633	5	0,664
70	0,714	7	0,664

Tabela 22 – Classificador *KNN* para potência

Na Tabela 23 observa-se o *score* do classificador para energia. O melhor *score* se encontra no *train split* de 30% e 50% com o valor de 0,447 que pode ser interpretado como 47,7% de acurácia.

train split (%)	score	cross validation	scores
30	0,388	3	0,385
50	0,477	5	0,391
70	0,428	7	0,363

Tabela 23 – Classificador *KNN* para energia

3.4.2.5 Classificador Bagging

A Figura 23 ilustra a implementação do treinamento da amostra para o classificador *Bagging*. Esse classificador combina a previsão de vários algoritmos de decisão. Com o valor do *score* pode-se chegar na interpretação da acurácia.

```
from sklearn.ensemble import BaggingClassifier

bag_model = BaggingClassifier(
    base_estimator=DecisionTreeClassifier(),
    n_estimators=100,
    max_samples=0.8,
    oob_score=True,
    random_state=0
)
bag_model.fit(X_train, y_train)
bag_model.oob_score_

bag_model.score(X_test, y_test)
```

Figura 23 – Valor de previsão para o *Bagging* com validação cruzada

Na Figura 24 vê-se a implementação da validação cruzada. Passou-se os parâmetros para o classificador *Bagging* e o valor da validação cruzada. No final, combinando vários algoritmos de previsão, obtém-se a média dos vetores gerados da validação cruzada.

```
bag_model = BaggingClassifier(
    base_estimator=DecisionTreeClassifier(),
    n_estimators=100,
    max_samples=0.8,
    oob_score=True,
    random_state=0
)
scores = cross_val_score(bag_model, inputs, target, cv=7)
scores
```

Figura 24 – Valor de previsão para o *Bagging* com validação cruzada

Na Tabela 22 observa-se o *score* do classificador para a potência. O melhor *score* se encontra no *train split* de 30% com o valor de 0,644 que pode ser interpretado como 64,4% de acurácia.

train split (%)	score	cross validation	scores
30	0,644	3	0,557
50	0,582	5	0,625
70	0,600	7	0,607

Tabela 24 – Classificador *Bagging* para potência

Na Tabela 25 tem-se o *score* do classificador para energia. O melhor *score* se encontra no *train split* de 30% com o valor de 0,400 que pode ser interpretado como 40,0% de acurácia.

train split (%)	score	cross validation	scores
30	0,400	3	0,424
50	0,311	5	0,469
70	0,333	7	0,447

Tabela 25 – Classificador *Bagging* para energia

Na busca pelo algoritmo *RandomSubspace* na biblioteca *Scikit-learn*, o algoritmo encontrado foi *BaggingClassifier*. Por este motivo, o *RandomSubspace* não foi implementado em *Python*.

4

RESULTADOS E DISCUSSÕES

Nesta seção é apresentada a interpretação dos resultados obtidos neste trabalho. É verificado se a proposta foi atendida e se o objetivo geral e objetivos específicos foram alcançados. A proposta deste trabalho foi o uso de mineração de dados e aprendizagem de máquina para prever informações relevantes de comunidades rurais da amazônia que pudessem servir para previsões de planejamento energético. Os resultados obtidos através da técnica foram animadores e demonstrou que é possível fazer previsões que auxiliam no planejamento energético para comunidades não eletrificadas que possuam características similares.

4.1 Resultados no WEKA e Python

Este estudo avaliou o desempenho de vários métodos de aprendizagem de máquina para prever a potência instalada. O estudo foi realizado no WEKA e no *Python*, usando a biblioteca do *PANDAS* e *Scikit-learn* para aprendizagem de máquina supervisionada.

Potência instalada:

Os resultados do estudo mostraram que os métodos de aprendizagem de máquina apresentaram um desempenho semelhante no WEKA e no *Python* e podem ser vistos na Tabela 26.

Método	Potencia instalada Weka(%)	Potência instalada Python (%)	Diferença de desempenho (%)
DecisionTree	68,8	79,2	10,4
Naive Bayes	68,5	75,9	7,1
OneRule	68,8	69,6	0,8
KNN	61,1	74,0	13,3
Bagging	68,8	64,4	-4,1

Tabela 26 – Diferemça de desempenho para potência instalada

O método *DecisionTree* obteve o melhor desempenho em ambas as ferramentas, com uma acurácia de 68,8% no Weka e 79,2% no Python. A diferença de desempenho entre 79,2 e 68,8 é de 10,4%. Isso significa que o *DecisionTree* com acurácia de 79,2% é capaz de prever corretamente 10,4% mais pontos do que o *DecisionTree* com acurácia de 68,8%.

Para o método *NaiveBayes* A diferença de desempenho entre 68,5 e 75,9 para NaiveBayes de potência instalada é de 7,4%. Isso significa que o *NaiveBayes* com acurácia de 75,9% é capaz de prever corretamente 7,4% mais pontos do que o *NaiveBayes* com acurácia de 65,9%.

Para o método *OneRule* a diferença de desempenho entre 68,8 e 69,6 de potência instalada é de 0,8%. Isso significa que o *OneRule* com acurácia de 69,6% é capaz de prever corretamente 0,8% mais pontos do que o *OneRule* com acurácia de 68,8%.

Para o método *KNN* a diferença de desempenho entre 61,1 e 74,4 para a potência instalada é de 13,3%. Isso significa que o *KNN* com acurácia de 74,4% é capaz de prever corretamente 13,3% mais pontos do que o KNN com acurácia de 61,1%. No entanto, é importante notar que a diferença de desempenho entre 74,4 e 61,1 é considerada significativa na literatura de aprendizagem de máquina. Isso ocorre porque a acurácia de 61,1% é considerada um desempenho médio para muitos problemas de previsão. A diferença de desempenho para o KNN é maior do que para os outros algoritmos. Isso ocorre porque o KNN é um algoritmo mais complexo que os outros algoritmos, os algoritmos mais complexos geralmente requerem mais dados para serem treinados e podem ser mais propensos ao *overfitting* que é quando o modelo se ajusta excessivamente aos dados de treinamento, ou seja, o modelo tem um desempenho excelente nos dados

de treino porém tem o desempenho ruim nos dados de teste.

Para o método *Bagging* a diferença de desempenho entre 68,5 e 64,4 para a potência instalada é de 4,1%. Isso significa que o *Bagging* com acurácia de 68,5% é capaz de prever corretamente 4,1% mais pontos do que o *Bagging* com acurácia de 64,4%.

A diferença de desempenho entre os algoritmos de aprendizagem de máquina utilizados para prever a potência instalada é pequena. Em todos os casos, os algoritmos são capazes de prever corretamente a potência instalada com uma precisão razoável.

Consumo de energia diário:

Os resultados do estudo mostraram que os métodos de aprendizagem de máquina apresentaram um desempenho semelhante no WEKA e no *Python* e podem ser vistos na Tabela 27.

Método	Potência instalada Weka(%)	Potência instalada Python (%)	Diferença de desempenho (%)
DecisionTree	68,5	61,4	7,1
Naive Bayes	46,3	25,9	20,4
OneRule	44,7	48,8	4,1
KNN	38,9	47,7	8,8
Bagging	53,7	64,4	10,7

Tabela 27 – Diferença de desempenho para consumo de energia diário

O método *DecisionTree* obteve o melhor desempenho em ambas as ferramentas, com uma acurácia de 68,5% no WEKA e 61,4% no *Python*. A diferença de desempenho entre as duas ferramentas foi de apenas 7.1%, o que também é considerado insignificante.

A diferença de desempenho entre 46,3 e 25,9 para *NaiveBayes* de consumo de energia diário é de 20,4%. Isso significa que o *NaiveBayes* com acurácia de 46,3% é capaz de prever corretamente 20,4% mais pontos do que o *NaiveBayes* com acurácia de 25,9%. No entanto, é importante notar que a diferença de desempenho entre 46,3 e 25,9 é considerada significativa na literatura de aprendizagem de máquina. Isso ocorre porque a acurácia de 25,9% é considerada um desempenho ruim para muitos problemas de previsão. Em geral, a diferença de desempenho entre 46,3 e 25,9 é grande. Em ambos os casos, os algoritmos são capazes de prever corretamente o consumo de energia diário,

mas o *NaiveBayes* com acurácia de 46,3% é significativamente mais preciso.

A diferença de desempenho entre 44,7 e 48,8 para *OneRule* de consumo de energia diário é de 4,1%. Isso significa que o OneRule com acurácia de 48,8% é capaz de prever corretamente 4,1% mais pontos do que o *OneRule* com acurácia de 44,7%.

A diferença de desempenho entre 38,9 e 47,7 para KNN de consumo de energia diário é de 8,8%. Isso significa que o KNN com acurácia de 47,7% é capaz de prever corretamente 8,8% mais pontos do que o KNN com acurácia de 38,9%.

A diferença de desempenho entre 53,7 e 64,4 para *Bagging* de consumo de energia diário é de 10,7%. Isso significa que o *Bagging* com acurácia de 53,7% é capaz de prever corretamente 10,7% mais pontos do que o *Bagging* com acurácia de 53,7%.

No geral diferença de desempenho entre os algoritmos é relativamente pequena. Entretanto, a maioria dos métodos teve maior desempenho com o uso da linguagem de programação Python com as bibliotecas *PANDAS* e *Scikit-Learn*. O algoritmo que mais se destacou nos dois cenários foi o *Decision Tree*. Isso significa que este algoritmo é a melhor opção para prever potência instalada.

5

Considerações Finais

5.1 Conclusão

Com base nos resultados obtidos, é possível concluir que os algoritmos de aprendizagem de máquina são uma ferramenta valiosa para prever a potência instalada e consumo de energia diário. No entanto, é importante escolher o algoritmo certo para o problema em questão. No caso deste trabalho, os objetivos propostos foram alcançados. Os algoritmos de aprendizagem de máquina foram capazes de prever a potência instalada com 68,5% e o consumo de energia com 79,2% de acurácia utilizando o método DecisionTree. Este método mostrou-se o melhor para o estudo de caso. Quanto as ferramentas utilizadas, o consumo de energia teve melhor acurácia com o uso do software Weka, enquanto o consumo de energia teve o melhor resultado com a ferramenta Python. Portanto, recomenda-se a adoção do método DecisionTree para prever os parâmetros de eletrificação de comunidades rurais, podendo tanto o Weka quanto o Python ser escolhidos para o desenvolvimento da solução, pois a diferença de acurácia, para o método, foi bem pequena.

Dimensionamento de sistemas de geração

Em regiões isoladas da Amazônia, o dimensionamento de sistemas de geração de energia é um desafio. Isso ocorre porque as demandas de energia são geralmente baixas e flutuantes, e as fontes de energia disponíveis são limitadas.

As previsões de potência instalada podem ser usadas para ajudar a resolver esse desafio. As previsões podem fornecer aos gestores de sistemas elétricos uma visão sobre as futuras demandas de energia, permitindo que eles planejem para adicionar novos geradores ou aumentar a capacidade dos geradores existentes.

Por exemplo, se as previsões indicam que a demanda de energia aumentará no futuro, os gestores de sistemas elétricos podem planejar para adicionar novos geradores solares fotovoltaicos. Esses geradores são uma boa opção para regiões isoladas da Amazônia, pois são renováveis e não requerem os grandes investimentos de infraestrutura de transmissão necessários nas extensões das redes atuais, pois passam a caracterizar uma geração distribuída.

Dimensionamento de sistemas de distribuição

O dimensionamento de sistemas de distribuição de energia também é um desafio em regiões isoladas da Amazônia. Isso ocorre porque as redes elétricas são geralmente pequenas e de baixa tensão.

As previsões de consumo de energia podem ser usadas para ajudar a resolver esse desafio. As previsões podem fornecer aos gestores de sistemas elétricos uma visão sobre as futuras demandas de energia, permitindo que eles planejem para construir novas linhas de transmissão ou aumentar a capacidade das linhas existentes.

Por exemplo, se as previsões indicam que o consumo de energia aumentará em uma determinada área, os gestores de sistemas elétricos podem planejar para construir novas linhas de transmissão de baixa tensão. Essas linhas são mais baratas e fáceis de construir do que linhas de alta tensão.

5.2 Trabalhos futuros

Os questionários utilizados para montar a base de dados foram aplicados no ano de 2017. Atualmente, algumas dessas comunidades isoladas receberam o forne-

cimento de energia. A proposta de um trabalho futuro seria comparar as previsões feitas neste estudo de caso com o resultado real, revisitando as pessoas que forneceram as informações socieconômicas e avaliando se a previsão do consumo de energia foi realizada.

REFERÊNCIAS

ALPADIN, E. *Introdution to Machine Learning*. 4. ed.. ed. [S.l.]: The MIT Press, 2020. ISBN 9788543014432. 19

BHATT, G. D. Management strategies for individual knowledge and organizational knowledge. *Journal of Knowledge Management*, v. 2, n. 1, p. 31–39, 2002. Disponível em: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.475.5303&rep=rep1&type=pdf.> 30

BREIMAN, L. Bagging predictors. *March Learn*, 1996. Disponível em: <https://link.springer.com/article/10.1007/BF00058655#citeas>. 46

BROWNLEE, J. How to load csv machine learning data in weka. *Machine Learning Mastery*, 2020. Disponível em: <https://machinelearningmastery.com/load-csv-machine-learning-data-weka/>. 36

CHANDRASHEKAR, S.; RAMAKRISHNAN, R. Data cleaning for data mining: Concepts and techniques. *Association for Computing Machinery*, 2014. 31

CRISTIANINI, N.; SHAWE-TAYLOR, J. *Introduction to machine learning*. [S.l.]: Cambridge University Press, 2020. 40

FARIA, M. M. *Detecção de intrusões em redes de computadores com base nos algoritmos KNN, K-Means++ e J48*. 2016. Disponível em: <https://www.cc.faccamp.br/Dissertacoes/MauricioMendesFaria.pdf>. 44

GOOGLE. Colaboratory. 2023. Disponível em: <https://research.google.com/colaboratory/intl/pt-BR/faq.html>. 46

HAND, D. J.; MANNILA, H.; SMYTH, P. *Principles of data mining*. [S.l.]: Springer, 2001. 35

HASTIE, T. et al. *The elements of statistical learning*. 2nd ed. ed. [S.l.]: Springer, 2009. 40

HOPPE, A. et al. Wisdom - the blurry top of human cognition in the dikw-model? *7th conference of the European Society for Fuzzy Logic and Technology*, 2011. Disponível em: <https://www.researchgate.net/publication/269081562_Wisdom_-_the_blurry_top_of_human_cognition_in_the_DIKW-model>. 31

IBM. Introduction to machine learning. 2019. Disponível em: <https://developer.ibm.com/articles/introduction-to-machine-learning/>. 48

IBM. Como funciona o aprendizado supervisionado? 2023. Disponível em: <https://www.ibm.com/br-pt/topics/supervised-learning>. 19

IBM. O que é aprendizado supervisionado? 2023. Disponível em: <https://www.ibm.com/br-pt/topics/supervised-learning>. 27

IEMA. *Eletrificação rural na Amazônia: desafios e oportunidades*. [S.l.]: Instituto de Energia e Meio Ambiente, 2022. 24

KINNEY, W. *Python for Data Analysis*. [S.l.]: Jupyter, 2017. 48

KUNCHEVA, L. I. *Combining Pattern Classifier-Methods and Algorithms*. 2. ed.. ed. [S.l.]: WILEY, 2014. 45

LIEW, A. Dikiw: Data, information, knowledge, intelligence, wisdom and their interrelationships. *Business Management Dynamics*, v. 2, n. 10, p. 49–62, 2013. Disponível em: <https://www.researchgate.net/publication/236870996_DIKIW_Data_Information_Knowledge_Intelligence_Wisdom_and_their_Interrelationships>. 30

MARKOVIć, M.; BOSSART, M.; HODGE, B.-M. Machine learning for modern power distribution systems: Progress and perspectives. *Journal of Renewable and Sustainable Energy*, v. 15, n. 3, p. 032301, 06 2023. ISSN 1941-7012. Disponível em: <https://doi.org/10.1063/5.0147592>. 26

MURPHY, K. P. M. *Machine Learning: A Probabilistic Perspective*. [S.l.]: The MIT Press, 2012. 20, 51

MüLLER, H.; FREYTAG, J.-C. Problems, methods, and challenges in comprehensive data cleansing. *Humboldt-Universität zu Berlin zu Berlin*, 2015. 29, 32

NADALIN R., . C. R. *A ciência de dados: uma abordagem prática*. [S.l.]: Novatec, 2023. 20

OLIVEIRA, A. C. *Máquina de Aprendizagem Mínima com Opção de Rejeição*. 2016. Disponível em: <https://repositorio.ufc.br/bitstream/riufc/52663/1/2016_acoliveira.pdf>. 44

ORACLE. O que é ia? saiba mais sobre inteligência artificial. 2023. Disponível em: <https://www.oracle.com/br/artificial-intelligence/what-is-ai/>. 19

PEDREGOSA, F. Scikit-learn: Machine learning in python. *Machine Learning Research*, 2011. Disponível em: <https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf>. 48

PONCE, A. C. et al. *Inteligência Artificial - Uma Abordagem de Aprendizado de Máquina*. 2. ed. [S.l.]: LTC, 2021. 28

QUINLAN, J. *Programs for machine learning*. [S.l.]: Morgan Kaufmann Publishers Inc, 1993. 42

RAHM, E.; DO, H. H. Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, v. 23, n. 4, p. 3–13, 2000. Disponível em: <http://sites.computer.org/debull/A00DEC-CD.pdf>. 32

RIBEIRO, A. P. *Dados, Informação e Conhecimento*. [S.l.]: UFMG - Universidade Federal de Minas Gerais, 2010. 30

TRINDADE, A. B. et al. Ciência de dados aplicada a questionários coletados de comunidade não eletrificadas do baixo rio negro no amazonas. *XIII Congresso Brasileiro de Planejamento Energético CBPE*, 2022. Disponível em: <https://www.researchgate.net/publication/374091654_Ciencia_de_dados_aplicada_a_questionarios_coletados_de_comunidade_nao_eletrificadas_do_baixo_Rio_Negro_no_Amazonas>. 19

TZIRAKIS, P.; TJORTJIS, C. T3c: improving a decision tree classification algorithm's interval splits on continuous attributes. *Advances in Data Analysis and Classification*, v. 11, 04 2016. 42

VASCO, D. Identificação de anomalias contextuais. *Porto: Universidade do Porto*, 2013. 32

WAIKATO, U. of. Weka 3: Machine learning software in java. 2023. Disponível em: <https://www.cs.waikato.ac.nz/ml/index.html>. 36, 38

WANG, X. Machine learning applications in power systems. 2020. Disponível em: <https://scholar.smu.edu/engineering_electrical_etds/39/>. 26

WITTEN, I. H.; FRANK, E.; HALL, M. A. *Data mining with WEKA: practical machine learning tools and techniques*. [S.l.]: Morgan Kaufmann, 2016. 39

XIE, J.; ALVAREZ-FERNANDEZ, I.; SUN, W. A review of machine learning applications in power system resilience. In: *2020 IEEE Power Energy Society General Meeting (PESGM)*. [S.l.: s.n.], 2020. p. 1–5. 26

6

ANEXOS

6.1 Anexo A

QUESTIONÁRIO

Projeto STAR Energy: **Sustainable and Replicable Off-grid Renewable Energy System for Riverside Communities in the Amazon, Brazil**

Questionnaire Q1 – Householder

Community name: _____
Researcher: _____
Day and time of visit: _____

LOT number / identification: _____ ( ) No identification
Resident: ( ) Permanent ( ) Eventual (How many days per month? _____)

If the questionnaire could not be applied: ( ) Closed ( ) Abandoned ( ) Would not respond
GPS Coordinates: S_____° _____._____ W_____° _____._____'

Parte I. IDENTIFICATION

Complete Name: _____

1. Gender ( ) M ( ) F

2. Age
( ) 15 - 25
( ) 26 - 36
( ) 37 - 45
( ) 46 - 55
( ) 56 or more

3. Civil Status
( ) Single
( ) Married
( ) Other union
( ) Separated

4. Education
( ) Does not know how to read
( ) Can read and write
( ) Completed elementary school
( ) Completed high school
( ) Completed the upper course
( ) incomplete elementary school
( ) Incomplete high school

5. Time of residence in the locality
Always lived () More than 10 years: _____ Less than 10 years: _____

6. Is there desire to relocate? ( ) Sim ( ) Não

7. If so, what is the reason _____

Part II. IDENTIFICATION OF THE PROPERTY

Note: Take an external photograph of the residence.

8. House
( ) Own
( ) Rented
( ) Lives together with another family
( ) Other _____

9. Number of rooms with and without electricity (if external, specify)

With Electricity	Without Electricity
( )	( ) Room
( )	( ) bedroom
( )	( ) Kitchen
( )	( ) bathroom
( )	( ) Other:_____

10. Housing conditions

( ) Wood ( ) Mixed
( ) Masonry ( ) Other

11. Type of material used in the cover
( ) Zinc ( ) Straw
( ) Clay tile ( ) Asbestos tile
( ) Other_____

12. State of conservation:
( ) Needing repairs ( ) Good ( ) Excellent

Part III. FAMILY HOUSING STRUCTURE LIVING IN THE HOUSING AND ITS EDUCATION

Number of people		M	F	M	F	
		In the house		At School		
How many until 5 years old?	Q13					Q19
How many from 6 to 10?	Q14					Q20
How many from 11 to 15?	Q15					Q21
How many from 16 to 20?	Q16					Q22
How many from 21 to 40?	Q17					Q23
How many over 40 years old?	Q18					Q24

25. Is there a school-age child (between 5 and 13 years old) who does not go to school?
( ) Yes ( ) No
26. What is the reason? _____

Part IV. PRODUCTIVE ACTIVITIES
27. What productive activities are carried out? _____
_____

28. Which is the main one? _____
29. How many contribute to the family's monthly income? _____
30. What is the family's monthly income? (Brazil´s Minimum Wage R$937.00)
( ) Up to one MW ( ) Between 5 and 10 MW
( ) Between 1 and 2 MW ( ) Between 10 and 20 MW
( ) Between 2 and 5 MW ( ) Above 20 MW

31. Family has breeding animals? Which are?
_____
32. Does the family have Plantation? (vegetables, grains or roots - beans, cassava, etc.)
_____
33. Does the family have fruit trees?
_____

34. What electrical equipment is used to develop work activity?
_____
35. What are the difficulties in carrying out the work? _____
_____
36. Is the production marketed? ( ) Yes No
37. If so, in what way? _____

Parte V. COOKING
38. How many meals a day? _____
39. How many people eat at home daily? _____
40. Where do you normally cook? () No Kitchen () Outdoor Kitchen () Indoor
41. Do you use a chimney or other exhaust to remove the smoke from the environment?
( ) Yes ( )No
42. What kind of stove do you use to prepare main meals? _____
43. How was this stove bought (won, bought in some city, built)?

44. How much was paid or spent on this stove? _____

PART VI. HEALTH

45. In the last month, did any resident of the residence suffer from any respiratory disease?
( ) No ( ) Did not know to say ( ) Did not want to inform () Yes (which disease? _____)

46. In the last month, has any resident of the residence suffered eye irritation or another vision problem?
( ) No ( ) Did not know to say ( ) Did not want to inform () Yes (which disease? _____)

47. In the last month, did any residents live under any kind of burn resulting from making food, using heated equipment or handling fuel?
( ) No ( ) Did not know to say ( ) Did not want to inform () Yes (details? _____)

48. In the last month, has any resident of the residence suffered from diarrhea?
( ) No ( ) Did not know to say ( ) Did not want to inform () Yes

PART VII. NEEDS AND ASPIRATIONS

49. What are the most important needs for you today? (place in order of preference starting with the number 1, going to the 10 and filling all the fields)
( ) Water (availability and / or potability)
( ) Sanitary conditions (bathrooms with correct waste disposal)
( ) Transport
( ) Electricity
( ) Housing
( ) Employment and permanent work
( ) Program to generate income for the community
( ) Health posts
( ) School / Education
( ) Protection against floods / storms

PARTE VIII. ENERGETIC STUDY

50. Which of the following energetics do you use?
( ) Firewood ( ) LPG ( )Charcoal ( ) 12V Battery ( ) A or AAA Battery ( ) B or C Battery
( ) D Battery ( ) Diesel Oil ( ) Gasoline ( ) Kerosene ( ) Candle ( ) Electricity ( ) Other

51. Qual a finalidade do uso do(s) energético(s) citado(s) ?
What is the purpose of the use of the cited energy?
Firewood: _____
LPG: _____
Coal: _____
12V battery: _____
A or AAA battery: _____
B or C battery: _____
D battery: _____
Diesel oil: _____
Gasoline: _____
Kerosene: (if lamparina (lamp), ask for details, the quantity and how many hours/day)
_____
Candle: _____
Electricity: _____

53. What is the monthly consumption of the above mentioned energy source (s)?
Firewood: _____ kg LPG: _____kg Coal: _____kg 12VBattery: _____Unit.
A or AAA battery: _____Unit. B or C battery: _____Unit. D battery: _____Unit.
Diesel Oil: ____Liter Gasoline: ____Liter Kerosene: ____Liter

Candle: _____Unit. Electricity: _____kWh

53. What is the purchase price of the cited energy? Where to buy and how to transport
Firewood: _____
LPG: _____
Coal: _____
12V battery: _____
A or AAA battery: _____
B or C battery: _____
D battery: _____
Diesel oil: _____
Gasoline: _____
Kerosene: _____
Candle: _____
Electricity: _____

54. How much was spent on the main source of electricity last month?
( ) R$ _____ ( ) Do not know ( ) Does not have electricity

PART IX. ELECTRICAL DEMAND

55. Note the existing LOAD at the consumer unit (quantity and specification):
(OBS1: take into account only what is installed in place)
(OBS2: device powers will be defined from a standard table)

Lamps (quantity / type / power) _____
Radio / SOUND (quantity and type) _____
Television (quantity, size and type) _____
Satellite receiver (quantity) _____
Water pump (amount of power) _____
Refrigerator / Freezer (quantity and capacity in liters): _____
Fan (quantity): _____
Blender (quantity): _____
Air conditioner (quantity and power): _____
Other (type and quantity of each) _____

56. If you had 24/7 electricity, what additional equipments would be bought and used (quantity and specification)

Lamps (quantity / type / power) _____
Radio / SOUND (quantity and type) _____
Television (quantity, size and type) _____
Satellite receiver (quantity) _____
Water pump (amount of power) _____
Refrigerator / Freezer (quantity and capacity in liters): _____
Fan (quantity): _____
Blender (quantity): _____
Air conditioner (quantity and power): _____
Other (type and quantity of each) _____

Related with the following sources of energy:

(There is no right or wrong answer, the idea is to catch the opinion of the dweller)

	Household energy sources:	Grid electricity | Firewood | Charcoal & coal | LGP | Kerosene | Diesel | Other (specify)
57	In your opinion, is {sources} readily available in this community?	(Yes/No)
58	In your opinion, is {sources} expensive in this community?	(Yes/No)
59	In your opinion, does {sources} cause health problems?	(Yes/No)
60	In your opinion, is using {sources} safe in this community?	(Yes/No)

6.2 Anexo B

BANCO DE DADOS NOMINAL

Genero	Educacao	Vende_o_que_produz	Renda	Tipo_de_casa	Potencia_instalada
F	Completou o ensino fundamental	NA	Até 1 salário	Madeira	Até 1000W
F	Não completou o ensino fundamental	Sim	De 2 até 5 salários	Madeira	De 2000W até 3000W
M	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	De 1000W até 2000W
F	Completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
NA	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	De 2000W até 3000W
M	Não completou o ensino fundamental	Sim	De 2 até 5 salários	Madeira	Até 1000W
F	Não completou o ensino fundamental	Não	De 2 até 5 salários	Madeira	Até 1000W
F	Completou o ensino médio	Não	De 2 até 5 salários	Madeira	Até 1000W
F	Completou o ensino fundamental	NA	Até 1 salário	Madeira	De 1000W até 2000W
M	Não completou o ensino fundamental	Não	Até 1 salário	Misturado	De 1000W até 2000W
M	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	De 1000W até 2000W
M	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
F	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
F	Completou o ensino fundamental	Não	Até 1 salário	Misturado	De 2000W até 3000W
F	Não sabe ler	Não	Até 1 salário	Madeira	Até 1000W
M	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
M	Completou o ensino fundamental	NA	De 2 até 5 salários	Madeira	De 1000W até 2000W
M	Não completou o ensino fundamental	NA	Até 1 salário	Madeira	De 1000W até 2000W
M	Completou o ensino superior	NA	De 2 até 5 salários	Madeira	Até 1000W
F	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
M	Completou o ensino fundamental	NA	NA	Madeira	Até 1000W
Y	Completou o ensino fundamental	Não	Até 1 salário	Madeira	De 1000W até 2000W
M	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	De 1000W até 2000W
M	Completou o ensino fundamental	Sim	Até 1 salário	Madeira	De 2000W até 3000W
F	Completou o ensino superior	Não	Mais de 5 salários	Madeira	Até 1000W
F	Não completou o ensino fundamental	Sim	Até 1 salário	Misturado	Até 1000W
M	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
F	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
F	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
F	Completou o ensino médio	Sim	De 2 até 5 salários	Madeira	Até 1000W
F	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
M	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
F	Completou o ensino superior	Não	De 2 até 5 salários	Misturado	Até 1000W
F	Completou o ensino médio	Sim	Até 1 salário	Madeira	Até 1000W
F	Completou o ensino superior	Sim	Mais de 5 salários	Madeira	Até 1000W
F	Não completou o ensino fundamental	NA	De 2 até 5 salários	Madeira	Até 1000W
F	Completou o ensino médio	Não	Até 1 salário	Alvenaria	Até 1000W
F	Não completou o ensino fundamental	Não	De 2 até 5 salários	Madeira	Até 1000W
F	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
M	Completou o ensino fundamental	Não	De 2 até 5 salários	Alvenaria	De 1000W até 2000W
M	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	NA
F	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	NA
F	Não sabe ler	Sim	Até 1 salário	Madeira	Até 1000W
F	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
M	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
F	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
F	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
F	Completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
F	Completou o ensino médio	Não	Até 1 salário	Misturado	Até 1000W

F	Completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
M	Completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
F	Completou o ensino médio	Não	Até 1 salário	Madeira	Até 1000W
F	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
M	Completou o ensino médio	Não	Até 1 salário	Madeira	Até 1000W
F	Não completou o ensino fundamental	Não	NA	Madeira	Até 1000W
M	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
F	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	De 1000W até 2000W
F	Completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
F	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	Mais de 3000W
F	Completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
F	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
M	Não completou o ensino fundamental	Sim	NA	Madeira	Até 1000W
M	Não completou o ensino fundamental	Não	NA	Madeira	Até 1000W
F	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
M	Completou o ensino médio	Sim	De 2 até 5 salários	Madeira	Até 1000W
F	Completou o ensino médio	NA	De 2 até 5 salários	Madeira	Até 1000W
M	Não completou o ensino fundamental	NA	De 2 até 5 salários	Madeira	De 1000W até 2000W
M	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	De 1000W até 2000W
M	Não sabe ler	Não	Até 1 salário	Madeira	De 1000W até 2000W
F	Completou o ensino médio	Não	Até 1 salário	Madeira	Até 1000W
M	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
M	Não completou o ensino fundamental	Sim	De 2 até 5 salários	Madeira	Até 1000W
F	Completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
M	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
F	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
M	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	De 1000W até 2000W
M	Não completou o ensino fundamental	NA	NA	Madeira	De 1000W até 2000W
F	Completou o ensino fundamental	Sim	De 2 até 5 salários	Misturado	Até 1000W
M	NA	NA	De 2 até 5 salários	Madeira	Até 1000W
F	Completou o ensino médio	Não	Mais de 5 salários	Alvenaria	NA
F	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	De 2000W até 3000W
M	Não completou o ensino fundamental	Sim	De 2 até 5 salários	Madeira	De 1000W até 2000W
F	Completou o ensino médio	NA	NA	Madeira	Até 1000W
F	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
M	Completou o ensino fundamental	Não	Até 1 salário	Misturado	Mais de 3000W
M	Completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
F	Completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
F	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
F	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	Mais de 3000W
F	Não completou o ensino fundamental	Sim	NA	Madeira	De 2000W até 3000W
F	Completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
M	Completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
F	Completou o ensino fundamental	Não	Até 1 salário	Madeira	De 2000W até 3000W
	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
M	Completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
F	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
F	Completou o ensino médio	Sim	NA	Madeira	De 1000W até 2000W
F	Completou o ensino médio	Não	Até 1 salário	Madeira	Até 1000W
M	Não sabe ler	Não	Até 1 salário	Madeira	Até 1000W

F	Completou o ensino médio	Não	Até 1 salário	Madeira	Até 1000W
M	Completou o ensino médio	Não	Até 1 salário	Madeira	De 1000W até 2000W
M	Completou o ensino médio	Sim	Até 1 salário	Madeira	De 1000W até 2000W
M	Completou o ensino fundamental	Sim	Até 1 salário	Madeira	De 2000W até 3000W
F	Completou o ensino médio	Sim	Até 1 salário	Madeira	Até 1000W
F	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
M	Não completou o ensino fundamental	Sim	NA	Madeira	De 1000W até 2000W
M	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
M	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
F	Completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
M	Completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
M	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
F	Não sabe ler	Sim	Até 1 salário	Madeira	De 1000W até 2000W
M	Não completou o ensino fundamental	Sim	De 2 até 5 salários	Madeira	De 2000W até 3000W
F	Completou o ensino fundamental	Sim	De 2 até 5 salários	Madeira	Até 1000W
F	Não sabe ler	Sim	NA	Madeira	Até 1000W
M	Não sabe ler	Sim	Até 1 salário	Madeira	Até 1000W
M	Completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
M	Completou o ensino médio	Sim	Até 1 salário	Alvenaria	Até 1000W
F	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
M	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
M	Não completou o ensino fundamental	NA	De 2 até 5 salários	Madeira	Até 1000W
F	Completou o ensino fundamental	Não	Até 1 salário	Madeira	De 1000W até 2000W
M	NA	Não	Até 1 salário	Madeira	De 1000W até 2000W
F	Completou o ensino médio	Não	Até 1 salário	Madeira	Mais de 3000W
M	Completou o ensino médio	NA	NA	Madeira	Até 1000W
F	NA	NA	NA	NA	Até 1000W
M	Completou o ensino superior	Não	De 2 até 5 salários	Misturado	De 2000W até 3000W
M	Completou o ensino médio	Sim	Até 1 salário	Madeira	NA
F	Completou o ensino fundamental	Não	Até 1 salário	Misturado	Até 1000W
F	Completou o ensino médio	Não	De 2 até 5 salários	Alvenaria	Até 1000W
M	Completou o ensino médio	Sim	De 2 até 5 salários	Misturado	Mais de 3000W
M	Completou o ensino fundamental	Não	Até 1 salário	Misturado	Até 1000W
F	Não completou o ensino fundamental	Sim	Até 1 salário	Misturado	Até 1000W
F	Não completou o ensino fundamental	Sim	Até 1 salário	Misturado	De 1000W até 2000W
F	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
M	Completou o ensino médio	Não	Até 1 salário	Madeira	Até 1000W
M	Não completou o ensino fundamental	Sim	De 2 até 5 salários	Madeira	Até 1000W
M	Completou o ensino fundamental	Sim	Até 1 salário	Alvenaria	Até 1000W
M	Não completou o ensino fundamental	Não	Até 1 salário	Misturado	Até 1000W
F	Completou o ensino médio	Sim	De 2 até 5 salários	Madeira	Até 1000W
M	Completou o ensino médio	Sim	De 2 até 5 salários	Misturado	Até 1000W
M	Completou o ensino médio	Sim	Até 1 salário	Madeira	Até 1000W
M	Completou o ensino médio	Não	Até 1 salário	Madeira	Até 1000W
M	Completou o ensino médio	Sim	Até 1 salário	Madeira	Até 1000W
F	Não completou o ensino fundamental	Não	Até 1 salário	Alvenaria	De 1000W até 2000W
M	Completou o ensino médio	Sim	Até 1 salário	Madeira	Até 1000W
F	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
F	Não completou o ensino fundamental	Não	De 2 até 5 salários	Madeira	Até 1000W
M	Não completou o ensino fundamental	Sim	De 2 até 5 salários	Madeira	De 1000W até 2000W

M	Completou o ensino superior	Não	NA	Madeira	Até 1000W
F	Completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
M	Completou o ensino fundamental	Sim	Até 1 salário	Madeira	De 1000W até 2000W
M	Completou o ensino médio	Sim	Até 1 salário	Madeira	Até 1000W
F	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	De 2000W até 3000W
F	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
F	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	NA
M	Não completou o ensino fundamental	Não	Até 1 salário	Madeira	Até 1000W
F	Completou o ensino fundamental	Não	NA	Madeira	De 2000W até 3000W
F	Não completou o ensino fundamental	Não	NA	Misturado	Até 1000W
M	Completou o ensino médio	Não	De 2 até 5 salários	Misturado	De 1000W até 2000W
F	Completou o ensino médio	Sim	NA	Misturado	Mais de 3000W
M	Completou o ensino fundamental	Não	Até 1 salário	Misturado	Até 1000W
M	Não completou o ensino fundamental	Sim	Até 1 salário	Madeira	Até 1000W
M	Completou o ensino médio	Sim	De 2 até 5 salários	Madeira	De 1000W até 2000W
M	Não completou o ensino fundamental	Não	Até 1 salário	Misturado	Até 1000W
M	Completou o ensino médio	Não	De 2 até 5 salários	Madeira	De 2000W até 3000W
M	Completou o ensino médio	Sim	Até 1 salário	Misturado	Até 1000W
M	Completou o ensino médio	Sim	Até 1 salário	Madeira	Mais de 3000W
F	Completou o ensino fundamental	Sim	Até 1 salário	Misturado	De 1000W até 2000W
M	Completou o ensino médio	Sim	De 2 até 5 salários	Alvenaria	De 1000W até 2000W
M	Não completou o ensino fundamental	Sim	De 2 até 5 salários	Misturado	De 1000W até 2000W
M	Completou o ensino fundamental	Não	Até 1 salário	Madeira	Mais de 3000W
F	Completou o ensino fundamental	NA	Mais de 5 salários	Madeira	De 1000W até 2000W
F	Completou o ensino fundamental	NA	Até 1 salário	Madeira	Até 1000W
M	Completou o ensino superior	NA	Mais de 5 salários	Misturado	Mais de 3000W
F	Completou o ensino fundamental	NA	Até 1 salário	Madeira	Até 1000W
F	Completou o ensino fundamental	NA	Até 1 salário	Madeira	Até 1000W
M	Completou o ensino fundamental	NA	Até 1 salário	Madeira	De 1000W até 2000W
F	Completou o ensino médio	NA	Até 1 salário	Madeira	De 1000W até 2000W

6.3 Anexo C

BANCO DE DADOS ORDINAL

	Genero	Educacao	Vende_o_que_produz	Renda	Tipo_de_casa	Potencia_instalada
0	0	0	1	0	1	0
1	0	3	1	1	1	2
2	1	3	1	0	1	1
3	0	0	0	0	1	0
4	0	3	0	0	1	2
5	1	3	1	1	1	0
6	0	3	0	1	1	0
7	0	1	0	1	1	0
8	0	0	1	0	1	1
9	1	3	0	0	2	1
10	1	3	1	0	1	1
11	1	3	0	0	1	0
12	0	3	1	0	1	0
13	0	0	0	0	2	2
14	0	4	0	0	1	0
15	1	3	0	0	1	0
16	1	0	1	1	1	1
17	1	3	1	0	1	1
18	1	2	1	1	1	0
19	0	3	0	0	1	0
20	1	0	1	0	1	0
21	2	0	0	0	1	1
22	1	3	1	0	1	1
23	1	0	1	0	1	2
24	0	2	0	2	1	0
25	0	3	1	0	2	0
26	1	3	0	0	1	0
27	0	3	1	0	1	0
28	0	3	1	0	1	0
29	0	1	1	1	1	0
30	0	3	0	0	1	0
31	1	3	0	0	1	0
32	0	2	0	1	2	0
33	0	1	1	0	1	0
34	0	2	1	2	1	0
35	0	3	1	1	1	0
36	0	1	0	0	0	0
37	0	3	0	1	1	0
38	0	3	0	0	1	0

39	1	0	0	1	0	1
40	1	3	0	0	1	0
41	0	3	1	0	1	0
42	0	4	1	0	1	0
43	0	3	0	0	1	0
44	1	3	1	0	1	0
45	0	3	1	0	1	0
46	0	3	1	0	1	0
47	0	0	1	0	1	0
48	0	1	0	0	2	0
49	0	0	0	0	1	0
50	1	0	0	0	1	0
51	0	1	0	0	1	0
52	0	3	0	0	1	0
53	1	1	0	0	1	0
54	0	3	0	0	1	0
55	1	3	0	0	1	0
56	0	3	1	0	1	1
57	0	0	1	0	1	0
58	0	3	1	0	1	3
59	0	0	1	0	1	0
60	0	3	1	0	1	0
61	1	3	1	0	1	0
62	1	3	0	0	1	0
63	0	3	1	0	1	0
64	1	1	1	1	1	0
65	0	1	1	1	1	0
66	1	3	1	1	1	1
67	1	3	1	0	1	1
68	1	4	0	0	1	1
69	0	1	0	0	1	0
70	1	3	0	0	1	0
71	1	3	1	1	1	0
72	0	0	0	0	1	0
73	1	3	1	0	1	0
74	0	3	1	0	1	0
75	1	3	0	0	1	1
76	1	3	1	0	1	1
77	0	0	1	1	2	0

78	1	3	1	1	1	0
79	0	1	0	2	0	0
80	0	3	1	0	1	2
81	1	3	1	1	1	1
82	0	1	1	0	1	0
83	0	3	0	0	1	0
84	1	0	0	0	2	3
85	1	0	1	0	1	0
86	0	0	1	0	1	0
87	0	3	0	0	1	0
88	0	3	1	0	1	3
89	0	3	1	0	1	2
90	0	0	1	0	1	0
91	1	0	0	0	1	0
92	0	0	0	0	1	2
93	0	3	0	0	1	0
94	1	0	0	0	1	0
95	0	3	0	0	1	0
96	0	1	1	0	1	1
97	0	1	0	0	1	0
98	1	4	0	0	1	0
99	0	1	0	0	1	0
100	1	1	0	0	1	1
101	1	1	1	0	1	1
102	1	0	1	0	1	2
103	0	1	1	0	1	0
104	0	3	1	0	1	0
105	1	3	1	0	1	1
106	1	3	0	0	1	0
107	1	3	0	0	1	0
108	0	0	1	0	1	0
109	1	0	1	0	1	0
110	1	3	1	0	1	0
111	0	4	1	0	1	1
112	1	3	1	1	1	2
113	0	0	1	1	1	0
114	0	4	1	0	1	0
115	1	4	1	0	1	0
116	1	0	1	0	1	0

117	1	1	1	0	0	0
118	0	3	1	0	1	0
119	1	3	0	0	1	0
120	1	3	1	1	1	0
121	0	0	0	0	1	1
122	1	3	0	0	1	1
123	0	1	0	0	1	3
124	1	1	1	0	1	0
125	0	3	1	0	1	0
126	1	2	0	1	2	2
127	1	1	1	0	1	0
128	0	0	0	0	2	0
129	0	1	0	1	0	0
130	1	1	1	1	2	3
131	1	0	0	0	2	0
132	0	3	1	0	2	0
133	0	3	1	0	2	1
134	0	3	0	0	1	0
135	1	1	0	0	1	0
136	1	3	1	1	1	0
137	1	0	1	0	0	0
138	1	3	0	0	2	0
139	0	1	1	1	1	0
140	1	1	1	1	2	0
141	1	1	1	0	1	0
142	1	1	0	0	1	0
143	1	1	1	0	1	0
144	0	3	0	0	0	1
145	1	1	1	0	1	0
146	0	3	0	0	1	0
147	0	3	0	1	1	0
148	1	3	1	1	1	1
149	1	2	0	0	1	0
150	0	0	1	0	1	0
151	1	0	1	0	1	1
152	1	1	1	0	1	0
153	0	3	0	0	1	2
154	0	3	1	0	1	0
155	0	3	1	0	1	0

156	1	3	0	0	1	0
157	0	0	0	0	1	2
158	0	3	0	0	2	0
159	1	1	0	1	2	1
160	0	1	1	0	2	3
161	1	0	0	0	2	0
162	1	3	1	0	1	0
163	1	1	1	1	1	1
164	1	3	0	0	2	0
165	1	1	0	1	1	2
166	1	1	1	0	2	0
167	1	1	1	0	1	3
168	0	0	1	0	2	1
169	1	1	1	1	0	1
170	1	3	1	1	2	1
171	1	0	0	0	1	3
172	0	0	1	2	1	1
173	0	0	1	0	1	0
174	1	2	1	2	2	3
175	0	0	1	0	1	0
176	0	0	1	0	1	0
177	1	0	1	0	1	1
178	0	1	1	0	1	1