Mineração de Subgrupos de Baterias de Íon-Lítion no Contexto de Predição de Tempo Vida Útil

Davi Lage[1], Eduardo Mio[1], João Pedro Fernandes[1], Leandro Diniz[1], Pedro Cimini[1]

[1]Departamento de Ciência da Computação – Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte, MG – Brasil

{davi.lage,eduardomio,joaofernandes,leandrodiniz,pedro.cimini}@dcc.ufmg.br

Abstract. *With the popularization of lithium-ion batteries, which are widely used in consumer electronics, electric vehicles, and aerospace applications, the interest in prediction models for the lifespan of this type of battery has also been increasing remarkably. However, regarding descriptive models, recent literature lacks references. In this context, this work investigates the topic of subgroup discovery in a labeled dataset of batteries and their respective lifespans. Subgroups were mined using four different approaches: SD-Map, SSD++, Apriori-SD, and Cortana. Additionally, a predictive model using the XGBoost algorithm was employed to validate the quality of the discovered subgroups. The work demonstrates the identification of significant patterns that can aid in the prediction and optimization of the lifespan of lithium-ion batteries.*

Resumo. *Com a popularização de baterias de Íon-Lítio, sendo amplamente utilizadas em eletrônicos de consumo, veículos elétricos e aplicações aeroespaciais, o interesse em modelos de predição no tempo de vida útil deste tipo de bateria também vem aumentando de maneira supreendente. Porém, tratando-se de modelos descritivos, a literatura recente carece de referências. Neste contexto, este trabalho investiga o tema de descoberta de subgrupos em uma base de dados rotulados de baterias e seus respectivos tempos de vida útil. Foram minerados subgrupos utilizando quatro abordagens diferentes: SD-Map, SSD++, Apriori-SD e Cortana. Além disso, foi utilizado um modelo preditivo com o algoritmo XGBoost para validar a qualidade dos subgrupos descobertos. O trabalho mostra que a identificação de padrões significativos que podem ajudar na previsão e otimização do tempo de vida útil de baterias de íon-lítio.*

1. Introdução

Baterias recarregáveis têm um papel extremamente importante na sociedade moderna. Especificamente, baterias recarregáveis de íon-lítio têm sido usadas em uma vasta gama de aplicações diferentes, como em eletrônicos de consumo (smartphones e notebooks), veículos elétricos, aparelhos de aeronáutica (satélites e VANTs), dentre outros [Goodenough and Park 2013]. Com a popularização de baterias de íon-lítio devido a queda do custo de materiais de produção e pelo alto tempo de vida deste tipo de bateria [Schmuch et al. 2018], soluções que visam prever o tempo de vida útil dessas baterias se tornam cada vez mais interessantes.

Neste contexto, podemos caracterizar os modelos de predição de tempo de vida de baterias em modelos empíricos ou modelos baseados em dados [Severson et al. 2019].

Modelos empíricos são construídos a partir do entendimento das reações químicas e físicas que ocorrem dentro das baterias, baseando-se em equações matemáticas derivadas de leis físicas conhecidas e geralmente requerem conhecimento prévio dos mecanismos de degradação específicos da bateria. Já modelos baseados em dados, utilizam-se de técnicas estatísticas para realizar a predição do tempo de vída útil das baterias. Uma vantagem deste tipo de técnica é que não há a necessidade de um entendimento detalhado dos processos físicos que ocorrem no processo de carga e descarga das baterias, encorajando mais profissionais a se interessarem pelo tema.

Porém, enquanto muitos dos esforços de pesquisa recentes são focados no desenvolvimento de novos modelos de predição [Severson et al. 2019, Fei et al. 2021, Che et al. 2022], não existem trabalhos na literatura, de conhecimento do grupo, que realizam um estudos de modelos descritivos no âmbito de baterias de íon-lítio. Uma abordagem comum para agregar conhecimentos em bases de dados é a de descoberta de subgrupos [Wrobel 1997]. O arcabouço de descoberta de subgrupos pode ser definido como uma técnica que visa descobrir relações interessantes entre diferentes objetos em relação à uma variável alvo, gerando padrões caracterizados por regras e subgrupos[Herrera et al. 2011]. Dessa forma, este trabalho descreve a aplicação de diferentes técnicas de descoberta de subgrupos em uma base de dados de baterias de íon-lítio, especificamente foram aplicados 4 abordagens diferentes em descobertas de subgrupos: SD-Map[Atzmueller and Puppe 2006], SSD++, Apriori-SD[Kavšek and Lavrač 2006] e o Cortana[Cor 2024]. Para validar a qualidade dos grupos descobertos, foi utilizado um modelo preditivo utilizando o algoritmo XGBoost[Chen and Guestrin 2016]. O repositorio do github com o código fonte do projeto pode ser acessado em: https://github.com/EduMio/TP-AD.

Assim, este estudo está organizado da seguinte maneira: na Seção 2 são estudos relacionados na área, mostrando dois trabalhos de modelos preditivos; na Seção 3 é descrita a base de dados que foi utilizada para a mineração dos subgrupos; na Seção 4 são descritos os passos de pré-processamento, falando sobre a remoção de outliers, normalização e discretização de dados; a Seção 5 apresenta com detalhes os algoritmos utilizados na descoberta de subgrupos, além do método de avaliação da qualidade dos subgrupos; na Seção 6 são apresentados os resultados de experimentação e por fim, na Seção 7, são apresentadas as conclusões do estudo.

2. Trabalhos Relacionados

Como mencionado na seção de introdução, não é de conhecimento do grupo algum trabalho na literatura corrente que realize a análise descritiva de dados realcionados a baterias de veículos elétricos.

Porém, tratando-se de modelos baseados em dados, a fim de predizer o tempo de vida útil de baterias, o trabalho de [Severson et al. 2019] é amplamente considerado um dos mais imporantes da área. Neste trabalho, são utilizados modelos de regressão *Elastic Net*[Zou and Hastie 2005] para realizar a predição do tempo de vida útil das baterias. São descritos 3 modelos diferentes, variando a quantidade de features dos modelos. Os autores mostram que os atributos gerados (todos utilizando apenas dados dos 100 primeiros ciclos das baterias) tem alto poder preditivo, com um modelo de apenas 1 *feature* obtendo desempenho satistfatório. Além de descrever o modelo de predição, os autores

disponibilizaram o dataset utilizado na pesquisa.

Outros trabalhos também abordam o problema da predição de tempo de vida utilizando *features* geradas nos primeiros 100 ciclos de vida útil. [Li et al. 2023] utiliza uma rede neural baseada em LSTM para realizar a predição do tempo de vida útil das baterias. Essa abordagem se mostrou eficaz, melhorando o desempenho que havia sido descrito anteriormente por [Severson et al. 2019].

[Kröger et al. 2023] aplicou o modelo de rede neural baseado em LSTM de [Li et al. 2023] em um cenário federado, mostrando que o treinamento em um ambiente distribuído é possível e pode ter consequências benéficas para aqueles que escolherem participar do ambiente colaborativo.

3. Base de dados

Foi utilizada a base de dados descrita no trabalho de [Severson et al. 2019]. A base de dados utilizada consiste em 124 baterias comerciais de baterias íon-lítio que foram cicladas até a falha em condições de carga rápida. Essas baterias de íon-lítio fosfato/grafita foram cicladas em cilindros horizontais. Esses ciclos consistem na repetida carga e descarga dessas baterias. Em cada ciclo, podemos observar a curva de descarga da bateria. Podemos perceber que, a cada ciclo, a vida útil de uma bateria, ou seja, a capacidade de carga total da bateria, se reduz. Quando é atingido o limiar de 80% da carga nominal de uma bateria, é definido que a bateria atingiu o limite da sua vida útil. Essa vida útil é caracterizada pelo número de ciclos necessários para atingir esse limiar. Esse número é chamado de *cycle life* da bateria.

A base de dados é dividida em 3 *batches*, definidos pela data de começo das medições realizadas na bateria. Os dados foram disponibilizados no formato *.mat* e, subsequentemente, transformados pelo grupo em arquivos *Pickle* para facilitar a sua utilização. A partir desse conjunto de dados temos acesso aos seguintes dados: o valor da resistência interna da bateria, a capacidade de carga/descarga, a temperatura média da bateria em um determinado ciclo, temperatura mínima/máxima de um dado ciclo, o tempo de carregamento de um dado ciclo, e todos os ciclos enumerados. Em relação a cada um dos ciclos individualmente, temos ainda os seguintes atributos:

- I - Corrente (A)
- t - Tempo
- T - Temperatura (ºC)
- V - Tensão (V)
- Q_c - Capacidade de carga (Ah)
- Q_d - Capacidade de descarga (Ah)
- Q_{dlin} - Capacidade de descarga linearmente interpolada
- $dQdV$ - Os vetores derivados da descarga
- $Tdlin$ - Temperatura linearmente interpolada

A partir desses dados, foram criadas uma série de *features* nas quais foram realizadas as análises. Essas features tem alto poder preditivo no contexto de tempo de vida útil das baterias [Severson et al. 2019]. São elas:

- `deltaQ_var` - A variância entre a capacidade de descarga linearmente interpolada no ciclo 100 e no ciclo 10

- `deltaQ_min` - O valor mínimo entre a capacidade de descarga linearmente interpolada no ciclo 100 e no ciclo 10
- `capFadeCycle2Slope` - A inclinação da curva de redução de capacidade entre os ciclos 2 e 100
- `capFadeCycle2Intercept` - A interceptação da curva de redução de capacidade entre os ciclos 2 e 100
- `qd2` - A capacidade de descarga no ciclo 2
- `avgChargeTime` - O tempo médio de recarga entre os ciclos 2 e 6
- `tempIntT` - A integral da temperatura entre os ciclos 2 e 100
- `minIR` - Valor mínimo da resistência interna entre os ciclos 2 e 100
- `IRDiff2And100` - A diferença da resistência interna dos ciclos 2 e 100
- `Cycle_life` - Número de ciclos totais da bateria

O foco nesses 100 primeiros ciclos de uma dada bateria se dá pelo conhecimento empírico de que esses ciclos têm uma correlação significativa com o número de ciclos totais de vida de uma dada bateria [Severson et al. 2019]. Esses dados, finalmente, foram colocados numa tabela de formato 124x10. Com essas *features*, foram realizadas análises com 6 diferentes algoritmos de descobertas de subgrupos, buscando encontrar subgrupos de *features* que possam explicar a duração, em número de ciclos, da vida de uma bateria.

4. Pré-processamento

Esta seção descreve os métodos utilizados para o pré-processamento de dados. Os métodos abordam a remoção de outliers, normalização dos dados, discretização e binarização de colunas. Estes métodos são detalhados abaixo.

4.1. Remoção de Outliers com Base no Qui-Quadrado

Removemos outliers da base de dados a partir da distância de Mahalanobis e o valor crítico do qui-quadrado. Primeiro, calcula-se o valor crítico do qui-quadrado para um determinado nível de significância, utilizando a fórmula:

$$\chi^2_{\text{crítico}} = \chi^2_{1-\alpha,k}$$

onde α é o nível de significância e k é o número de dimensões do DataFrame.

Em seguida, calcula-se a distância de Mahalanobis para cada linha do DataFrame. A distância de Mahalanobis é dada por:

$$D_M(\mathbf{x}) = \sqrt{(\mathbf{x} - \mu)^\top \mathbf{\Sigma}^{-1} (\mathbf{x} - \mu)}$$

onde $\mathbf{x}$ é o vetor de atributos da amostra, μ é o vetor média dos dados, e $\mathbf{\Sigma}$ é a matriz de covariância dos dados. Para cada linha, computa-se essa distância e compara-se com o valor crítico do qui-quadrado. Linhas cuja distância de Mahalanobis excede o valor crítico são consideradas outliers e removidas do DataFrame.

4.2. Normalização dos Dados

Normalizamos os dados para um intervalo de 0 a 100. A normalização é feita usando a técnica Min-Max Scaling, que transforma os dados da seguinte forma:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \times (b - a) + a$$

onde x é o valor original, x' é o valor normalizado, $x_{\min}$ e $x_{\max}$ são os valores mínimo e máximo do atributo, respectivamente, e $[a, b]$ é o intervalo desejado, neste caso, $[0, 100]$. Este processo garante que todos os valores dos atributos sejam escalonados para o intervalo especificado.

4.3. Discretização dos Dados com Limites Inferiores

Discretizamos os dados em um número especificado de bins, atribuindo a cada valor o limite inferior do bin ao qual pertence. A discretização é realizada da seguinte forma:

1. Determina-se os limites dos bins, dividindo o intervalo do atributo em partes iguais.

2. Cada valor é então mapeado para o limite inferior do bin correspondente.

Os limites dos bins são calculados usando a fórmula:

$$\text{edges} = \text{linspace}(x_{\min}, x_{\max}, n + 1)$$

onde n é o número de bins. Em seguida, cada valor é mapeado para o limite inferior do bin em que se encontra.

4.4. Binarização de Colunas

Convertemos as colunas do DataFrame em colunas binárias. Para cada valor único de uma coluna, cria-se uma nova coluna binária que indica a presença ou ausência desse valor. Este processo envolve: 1. Arredondar os valores da coluna para os inteiros mais próximos. 2. Para cada valor único na coluna, criar uma nova coluna binária que recebe 1 quando o valor arredondado é igual ao valor único e 0 caso contrário.

Este procedimento permite representar os dados originais de forma binária, facilitando a análise de padrões discretos nos dados.

5. Algoritmos Utilizados

A seção abaixo descreve os algoritmos que foram utilizados para a descoberta dos subgrupos, além do método de avaliação da qualidade dos subgrupos encotrados.

5.1. SSD++

Para a descoberta de subgrupos que mais excedem a média da variável de interesse *cycle_life*, utilizamos o algoritmo SSD++ em combinação com o *beam search*. O algoritmo SSD++ (Subgroup Set Discovery) é uma técnica avançada para a descoberta de subgrupos que visa encontrar conjuntos de subgrupos interessantes em dados complexos.

O processo do *beam search* começa com um conjunto inicial vazio e, iterativamente, expande os subgrupos adicionando atributos que melhoram o critério de qualidade escolhido. Em cada iteração, apenas um número fixo de subgrupos mais promissores (o *beam width*) é mantido para futuras expansões, limitando assim o espaço de busca e tornando o algoritmo mais eficiente.

Especificamente, para nossa análise, seguimos os seguintes passos:

1. Inicialização: Começamos com um conjunto inicial vazio de subgrupos, M, e calculamos a média inicial da variável de interesse *cycle_life*, denotada como μ_M.

2. Expansão dos Subgrupos Em cada iteração, os subgrupos existentes são expandidos adicionando novos atributos. Para cada expansão, calculamos a diferença (Δ) entre a média da variável *cycle_life* para o novo subgrupo e a média atual μ_M.

3. Seleção dos Subgrupos Promissores Apenas os subgrupos com a maior Δ positiva são mantidos. Este processo garante que continuamos a explorar apenas os subgrupos que melhoram a média da variável de interesse.

4. Atualização do Conjunto de Subgrupos Os subgrupos selecionados são adicionados ao conjunto M, e a média μ_M é atualizada para refletir a média dos subgrupos agora presentes em M.

5. Critério de Parada: O processo continua iterativamente até que um número máximo de iterações seja alcançado ou não haja mais subgrupos com Δ positiva.

A aplicação do SSD++ com *beam search* permite a descoberta eficiente de subgrupos que maximizam a média da variável de interesse, fornecendo insights valiosos sobre os padrões presentes nos dados.

5.2. SD-Map

Para a descoberta de subgrupos que mais excedem a média da variável de interesse *cycle_life*, utilizamos o algoritmo SD-Map. O SD-Map é um algoritmo eficiente que combina o método de mineração de padrões frequentes FP-Growth com técnicas de descoberta de subgrupos para identificar padrões interessantes nos dados. Para aumentar a diversidade dos nossos testes, utilizamos também o algoritmo Apriori no lugar do FP-Growth para avaliar se havia alguma mudança na análise, porém a explicação abaixo está correta tanto para o Apriori quanto para o FP-Growth.

O processo do SD-Map para a descoberta de subgrupos segue os seguintes passos:

1. Mineração de Padrões Frequentes O algoritmo começa utilizando o FP-Growth para minerar padrões frequentes nos dados. O FP-Growth é um algoritmo de mineração de padrões que constrói uma estrutura de árvore de padrões (FP-Tree) a partir do conjunto de dados e extrai padrões frequentes de maneira eficiente sem a necessidade de gerar candidatos explicitamente.

2. Cálculo da Qualidade dos Subgrupos: Para cada padrão frequente identificado pelo FP-Growth, calcula-se a média da variável de interesse *cycle_life* para os subgrupos correspondentes. A qualidade de um subgrupo é medida pela diferença (Δ) entre a média do subgrupo e a média geral dos dados. Formalmente, se μ_S é a média do subgrupo e μ_G é a média geral, então:

$$\Delta = \mu_S - \mu_G$$

3. Seleção dos Subgrupos Relevantes: Após calcular a qualidade para cada subgrupo, os subgrupos que mostram uma Δ positiva significativa são selecionados como subgrupos relevantes. Isso significa que apenas os subgrupos que excedem a média da variável de interesse de maneira substancial são considerados interessantes.

4. Iteração e Refinamento: O processo de mineração e seleção pode ser iterado e refinado para explorar subgrupos adicionais ou para ajustar os parâmetros do FP-Growth e do SD-Map, visando melhorar a qualidade dos subgrupos identificados.

5. Critério de Parada: O processo continua até que todos os padrões frequentes tenham sido avaliados ou até que um critério de parada predefinido seja alcançado, como um número máximo de subgrupos relevantes identificados ou um limite baseado na qualidade dos subgrupos.

O uso do SD-Map permite uma combinação poderosa de mineração de padrões frequentes e descoberta de subgrupos, identificando de forma eficiente os subgrupos que maximizam a média da variável de interesse *cycle_life*. Este método é especialmente útil para analisar grandes conjuntos de dados e descobrir padrões significativos que podem não ser aparentes com métodos de análise mais simples.

5.3. Apriori-SD

O Apriori-SD é um algoritmo projetado para descoberta de subgrupos, adaptando o aprendizado de regras de associação. Ele incorpora um esquema de ponderação de exemplos durante o pós-processamento das regras, utiliza uma função de qualidade de regra modificada que inclui pesos de exemplos no heurístico de precisão relativa ponderada e implementa um esquema de classificação probabilística.

O algoritmo começa gerando um conjunto inicial de regras usando o Apriori-C. As regras são ordenadas de acordo com a função de qualidade de precisão relativa ponderada. A melhor regra é selecionada e os exemplos cobertos por ela são reponderados. O procedimento se repete até que todos os exemplos tenham sido cobertos mais de um determinado número de vezes ou não existam mais regras no conjunto.

5.4. Cortana

O Cortana [Cor 2024] é uma ferramenta de mineração de dados projetada para descobrir padrões locais em conjuntos de dados. Ela inclui um algoritmo genérico de descoberta de subgrupos que pode ser configurado de várias maneiras para implementar diferentes formas de descoberta de padrões locais. A ferramenta é capaz de lidar com diversos tipos de dados, tanto para os atributos de entrada quanto para os atributos-alvo, incluindo dados nominais, numéricos e binários.

5.5. Construção do Modelo Preditivo com XGBoost e Avaliação de Importância com SHAP

Para construir um modelo preditivo da variável de interesse *cycle_life*, utilizamos o algoritmo XGBoost [Chen and Guestrin 2016], que é uma técnica avançada de boosting baseada em árvores de decisão. O processo foi realizado em várias etapas, conforme descrito a seguir.

Primeiro, realizamos a discretização dos dados originais. Os dados foram transformados em intervalos discretos, o que permitiu a aplicação de técnicas de mineração e modelagem de maneira mais eficiente. Após a discretização, treinamos o modelo preditivo XGBoost utilizando os atributos discretizados.

A importância de cada atributo no modelo preditivo foi avaliada utilizando os valores de Shapley (SHAP) [Shapley et al. 1953][Lundberg and Lee 2017]. Os valores de Shapley fornecem uma medida da contribuição de cada atributo para a predição do modelo. Calculamos a importância de cada atributo e definimos um ponto de corte de

importância baseado no valor da feature de menor importância significativa. Este ponto de corte serve como um limite abaixo do qual as features são consideradas não relevantes.

Depois de estabelecer o modelo preditivo e o ponto de corte de importância, realizamos uma nova fase de modelagem. Nesta fase, utilizamos um novo dataset composto de atributos binários que indicam se uma amostra pertence ou não a um subgrupo específico, conforme descoberto nos processos de mineração de subgrupos anteriores. Novamente, treinamos o modelo XGBoost com este novo dataset.

A importância dos subgrupos no novo modelo foi novamente avaliada utilizando os valores de Shapley. Subgrupos com importância acima do ponto de corte definido na primeira fase do modelo foram considerados relevantes. Isso permite a identificação dos subgrupos mais significativos para a predição da variável de interesse *cycle_life*.

Este método de duas fases, combinando a discretização dos dados, modelagem com XGBoost e avaliação de importância com SHAP, permite não apenas a construção de um modelo preditivo robusto, mas também a identificação das features e subgrupos mais relevantes para a variável de interesse. Isso é particularmente útil para entender a estrutura subjacente dos dados e para realizar análises mais detalhadas e interpretáveis.

6. Experimentação e Resultados

6.1. Configuração de experimentos

Os experimentos foram realizados utilizando implementações em python. Foi realizada uma breve experimentação para decidir os valores dos hiperparâmetros dos algoritmos implementados. Para os algoritmos SD-Map e Apriori-SD, em um primeiro momento, foi utilizado um valor de suporte mínimo de 1%. Porém, após o tratamento dos dados, o grupo chegou a conclusão que o valor de suporte de 5% estava gerando grupos de qualidade superior, portanto foi utilizado nos resultados finais. Para o algoritmo SSD++ foi tomado o valor de largura de beam de 150, devido ao tamanho pequeno da base dados. Para o Cortana, foi utilizada a medida de qualidade*Average*, para se aproximar dos outros algoritmos que utilizamos. O *target "cycle_life"* é o atributo principal em que estamos trabalhando para tentar encontrar subgrupos próximos das baterias. A profundidade 4 foi escolhida com base nos experimentos de [Meeng and Knobbe 2021] e a estratégia *best first* foi escolhida também devido ao tamanho pequeno da base de dados. A configuração do software pode ser vista na Figura 1.

6.2. Resultados

A tabela abaixo compara os três algoritmos de mineração de subgrupos utilizados. A análise comparativa revelou que, embora o SD-Map tenha encontrado um menor número de subgrupos, a importância média dos subgrupos relevantes foi a mais alta. O SSD++ encontrou o maior número de subgrupos, mas com uma importância média menor. Cortana apresentou um balanço entre o número de subgrupos relevantes e a sua importância média. Estes resultados demonstram a eficácia de cada método em diferentes contextos de mineração de subgrupos e a importância da escolha do algoritmo adequado para a análise de dados complexos.

7. Conclusão

O objetivo deste projeto é encontrar subgrupos de atributos que maximizam a média da variável alvo, número de ciclos de vida da bateria. Isso permite identificar quais faixas de

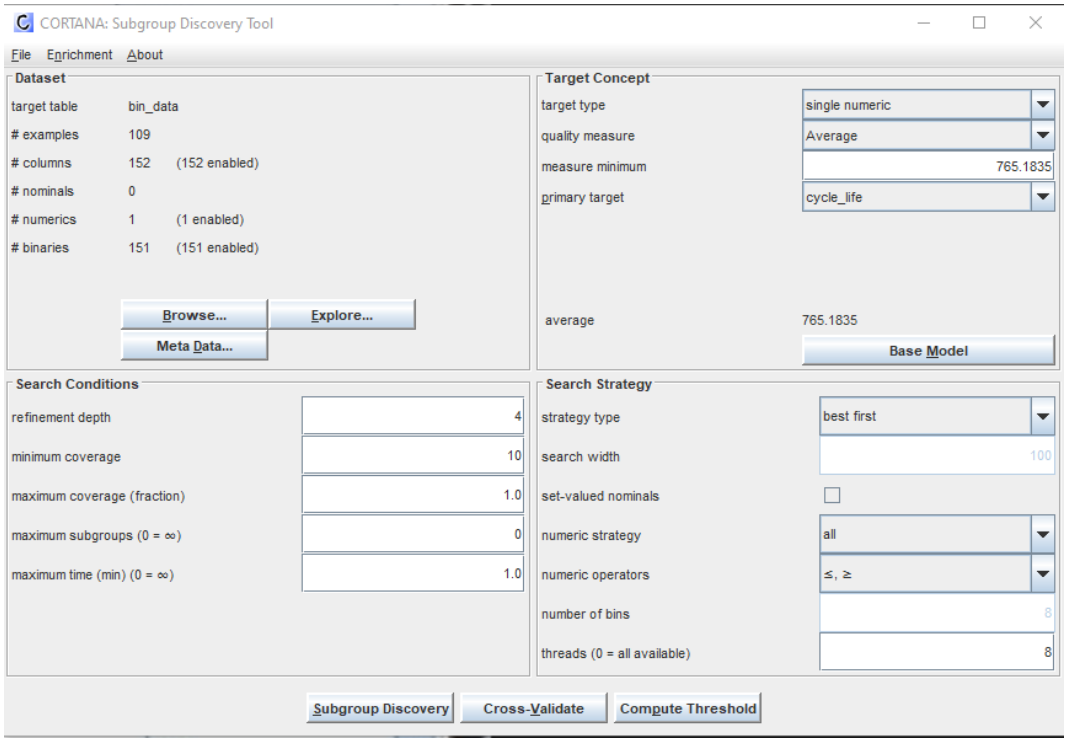

Figura 1. Configuração do Cortana

Algoritmo	Nº Minerados	Nº Relevantes	Importância Média
SD-Map	50	17	26.3
Apriori	50	17	26.3
SSD++	938	15	21.4
Cortana	937	28	18.03

Tabela 1. Comparação entre os Algoritmos de Mineração de Subgrupos

valores estão correlacionadas a uma maior expectativa de vida das amostras. Para isso, utilizamos quatro técnicas de mineração de subgrupos para avaliar quão bem são suas performances para o problema e validamos as soluções encontradas utilizando a significância preditiva de cada subgrupo para a variável alvo. Com isso, foi possível descobrir que o SD-Map, Apriori e SSD++ apresentam resultados semelhantes, com número de subgrupos descobertos e importância média próximos, já o Cortana encontra mais subgrupos relevantes, porém com menor importância média. Podemos concluir que cumprimos o objetivo de descoberta de subgrupos significativos, usando o argumento de que todos os subgrupos demarcados como importantes têm capacidade preditiva maior ou igual à feature de menor importância preditiva original.

Referências

(2024). Cortana framework. Disponível em: https://datamining.liacs.nl/cortana.html.

Atzmueller, M. and Puppe, F. (2006). Sd-map – a fast algorithm for exhaustive subgroup discovery. In Fürnkranz, J., Scheffer, T., and Spiliopoulou, M., editors, *Knowledge*

Discovery in Databases: PKDD 2006, pages 6–17, Berlin, Heidelberg. Springer Berlin Heidelberg.

Che, Y., Deng, Z., Tang, X., Lin, X., Nie, X., and Hu, X. (2022). Lifetime and aging degradation prognostics for lithium-ion battery packs based on a cell to pack method. *Chinese Journal of Mechanical Engineering*, 35(1):4.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Fei, Z., Yang, F., Tsui, K.-L., Li, L., and Zhang, Z. (2021). Early prediction of battery lifetime via a machine learning based framework. *Energy*, 225:120205.

Goodenough, J. B. and Park, K.-S. (2013). The li-ion rechargeable battery: A perspective. *Journal of the American Chemical Society*, 135(4):1167–1176. PMID: 23294028.

Herrera, F., Carmona, C. J., González, P., and del Jesus, M. J. (2011). An overview on subgroup discovery: foundations and applications. *Knowledge and Information Systems*, 29(3):495–525.

Kavšek, B. and Lavrač, N. (2006). Apriori-sd: Adapting association rule learning to subgroup discovery. *Applied Artificial Intelligence*, 20(7):543–583.

Kröger, T., Belnarsch, A., Bilfinger, P., Ratzke, W., and Lienkamp, M. (2023). Collaborative training of deep neural networks for the lithium-ion battery aging prediction with federated learning. *eTransportation*, 18:100294.

Li, X., Yu, D., Søren Byg, V., and Daniel Ioan, S. (2023). The development of machine learning-based remaining useful life prediction for lithium-ion batteries. *Journal of Energy Chemistry*, 82:103–121.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc.

Meeng, M. and Knobbe, A. (2021). For real: a thorough look at numeric attributes in subgroup discovery. *Data Mining and Knowledge Discovery*, 35(1):158–212.

Schmuch, R., Wagner, R., Hörpel, G., Placke, T., and Winter, M. (2018). Performance and cost of materials for lithium-based rechargeable automotive batteries. *Nature Energy*, 3(4):267–278.

Severson, K. A., Attia, P. M., Jin, N., Perkins, N., Jiang, B., Yang, Z., Chen, M. H., Aykol, M., Herring, P. K., Fraggedakis, D., Bazant, M. Z., Harris, S. J., Chueh, W. C., and Braatz, R. D. (2019). Data-driven prediction of battery cycle life before capacity degradation. *Nature Energy*, 4(5):383–391.

Shapley, L. S. et al. (1953). A value for n-person games.

Wrobel, S. (1997). An algorithm for multi-relational discovery of subgroups. In Komorowski, J. and Zytkow, J., editors, *Principles of Data Mining and Knowledge Discovery*, pages 78–87, Berlin, Heidelberg. Springer Berlin Heidelberg.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2):301–320.