

Análise de perdas da linha de Montagem em uma Indústria Automotiva

Loss analysis on the assembly line of an automobile industry

Kláudius Soares¹

 orcid.org/0009-0005-0585-1634

Marcelo Machado¹

 orcid.org/0000-0002-3267-1488

Vanessa Santos¹

 orcid.org/0009-0008-5605-2658

Adriano José¹

 orcid.org/0009-0004-6624-6385

¹ Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, Brasil. E-mail: kladius.soares@gmail.com

DOI: 10.25286/rep.v9i1.2782

Esta obra apresenta Licença Creative Commons Atribuição-Não Comercial 4.0 Internacional.

Como citar este artigo pela NBR 6023/2018: Kláudius Soares; Marcelo Machado; Vanessa Santos; Adriano José. Análise de perdas da linha de Montagem em uma Indústria Automotiva. Revista de Engenharia e Pesquisa Aplicada, v.9, n. 1, p. 79-85, 2024. DOI: 10.25286/rep.v9i1.2782

RESUMO

Este trabalho apresenta uma abordagem inovadora para a análise de perdas na linha de montagem de uma indústria automotiva, visando otimizar a eficiência do processo produtivo. A metodologia proposta utiliza o algoritmo de *machine learning* *XGBoost* em conjunto com técnicas de *feature engineering*. A análise de perdas é crucial para identificar e mitigar problemas que impactam a produtividade e qualidade na produção automotiva. O *XGBoost*, um algoritmo de aprendizado de máquina baseado em árvores de decisão, é escolhido devido à sua capacidade de lidar eficientemente com conjuntos de dados complexos e grandes volumes de informações. A fase de *feature engineering* desempenha um papel crucial na seleção de características relevantes para melhor capturar as nuances do processo de montagem. Os resultados experimentais demonstram a eficácia da abordagem proposta na detecção e prevenção de perdas na linha de montagem. O modelo *XGBoost*, treinado com dados historicamente relevantes, revela-se capaz de antecipar falhas potenciais, permitindo intervenções proativas para evitar interrupções no processo produtivo.

PALAVRAS-CHAVE: Machine Learning, XGBoost, Feature Engineering, Perdas, Processo, Automotivo.

ABSTRACT

This work presents an innovative approach to analyzing losses in the assembly line of an automotive industry, aiming to optimize the efficiency of the production process. The proposed methodology uses the XGBoost machine learning algorithm in conjunction with feature engineering techniques. Loss analysis is crucial to identify and mitigate problems that impact productivity and quality in automotive production. XGBoost, a decision tree-based machine learning algorithm, is chosen because of its ability to efficiently deal with complex data sets and large volumes of information. The feature engineering phase plays a crucial role in selecting relevant features to better capture the nuances of the assembly process. The experimental results demonstrate the effectiveness of the proposed approach in detecting and preventing losses on the assembly line. The XGBoost model, trained with historically relevant data, is capable of anticipating potential failures, allowing proactive interventions to avoid interruptions in the production process.

KEY-WORDS: Machine Learning, XGBoost, Feature Engineering, Losses, Process, Automotive.

1 INTRODUÇÃO

A análise de perdas em uma linha de montagem automotiva é um aspecto fundamental da gestão da produção e da qualidade na indústria automobilística [1]. Ela se concentra na identificação, medição e eliminação de desperdícios, ineficiências e problemas que podem ocorrer durante o processo de montagem de veículos [2]. Essas perdas podem ocorrer em várias etapas do processo de produção e têm um impacto significativo nos custos, na qualidade e na eficiência operacional de uma fábrica automotiva [2].

1.1 CONTEXTUALIZAÇÃO

No processo de fabricação de uma montadora de automóveis, várias etapas são necessárias para obter o produto final, incluindo a montagem de conjuntos elétricos, motores, assentos, pneus e outras peças. Cada etapa é realizada em estações de trabalho específicas ao longo da linha de produção. No entanto, diversos fatores podem impactar a execução das operações de montagem, tais como a falta de componentes, uso de materiais incorretos ou defeituosos, atividades executadas de forma inadequada pelos operadores, entre outros.

1.2 DESCRIÇÃO DO PROBLEMA

Em um processo produtivo ideal todas as etapas de trabalhos são executadas de acordo com o tempo padrão resultando num produto dentro do custo e qualidade planejados. Em um processo real ocorrem interrupções provenientes de ocorrências no processo que provocam perdas de tempo no ciclo produtivo. É essencial considerar que a soma dessas perdas pode ultrapassar o tempo planejado para a produção do veículo. Esse acúmulo de tempos de reação pode levar a paradas na linha e a perdas produtivas significativas.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Criar um modelo capaz de identificar áreas que tem maior impacto nas perdas de tempo na produção dos veículos.

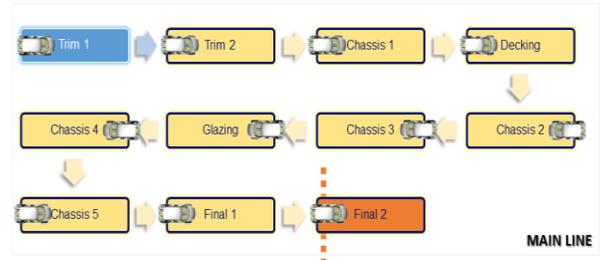
1.3.2 Objetivos Específicos

- Identificar os pontos de perdas mais significantes a partir de modelo de *machine*

learning gerado a partir de dados do processo produtivo;

- Analisar as variáveis do banco de dados mostrando a correlação entre elas, ajudando a identificar padrões, tendências e associações que podem ser úteis para tomada de decisões ou para entender melhor o comportamento das perdas.

Figura 1 – Exemplo linha de produção em indústria automotiva.



Fonte: Elaborado pelo autor.

1.4 JUSTIFICATIVA

Diante do crescente panorama de competitividade no setor automotivo brasileiro, as empresas do segmento de autopeças reconhecem a urgência tornar os processos mais competitivos. Essa iniciativa visa aprimorar seus sistemas produtivos e implementar análises de dados estratégicas, proporcionando assim embasamento crucial para a tomada de decisões assertivas na redução de perdas com mínimo investimento de recurso.

2 FUNDAMENTAÇÃO TEÓRICA

O OEE (*Overall Equipment Effectiveness*) é uma métrica que mede a eficiência global do equipamento, levando em consideração a disponibilidade, desempenho e qualidade [1,2]. É calculado multiplicando-se esses três fatores [2]. O OEE pode ser usado para identificar melhorias na produção e monitorar o desempenho ao longo do tempo. Pode ser aplicado em vários setores e usado com outras ferramentas de gerenciamento de produção, para melhorar ainda mais sua eficiência [2].

Big data refere-se a grandes volumes de dados coletados de várias fontes. É caracterizado por seu volume, velocidade, variedade e veracidade, tornando sua análise desafiadora. A análise de big data envolve técnicas avançadas para extrair informações valiosas desses conjuntos de dados.

Tecnologias como Hadoop, Spark e NoSQL são usadas nesse processo[3].

A aplicação de big data é amplamente utilizada em setores como finanças, saúde, varejo e manufatura. Ajuda as empresas a tomar decisões informadas, melhorar a eficiência, identificar tendências e fornecer melhores serviços aos clientes[4]. Em resumo, a análise de big data é uma técnica poderosa para extrair informações valiosas. [3,4].

2.1 PRÉ-PROCESSAMENTO E MINERAÇÃO DE DADOS

O pré-processamento de dados é uma etapa crucial na análise de dados, que envolve a preparação e tratamento dos dados brutos antes de aplicar técnicas de análise [5,7]. As etapas comuns de pré-processamento de dados incluem:

- Preparação, organização e estruturação dos dados. Ocorre antes da mineração de dados;
- Remoção dos campos indesejados;
- Conversão do arquivo;
- Seleção de período;
- Limpeza de campos vazios ou informações divergentes do padrão;

A mineração de dados é uma abordagem computacional empregada em análises para examinar extensos conjuntos de dados, utilizando ferramentas e métodos específicos. Por meio da mineração de dados, as organizações podem identificar padrões e relações previamente não evidentes em seus dados, transformando informações brutas em conhecimento aplicável [6,7].

As bases de dados utilizadas no projeto foram:

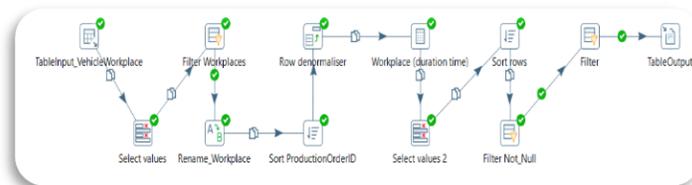
- **Vehicle Workplace:** Indica o momento de entrada/ saída dos veículos nas estações de trabalho.
- **Status List (Alarmes):** Lista de alarmes acionados e sua duração em cada etapa da linha de produção.

Etapas de pré-processamento e mineração do banco de dados *Vehicle Workplace*:

1. *File input (BD MES)*
2. *Select Values*
3. *Filter workplaces*
4. *Rename workplace*
5. *Sort ProductionOrderID*
6. *Row denormaliser*

7. *Workplace (duration time)*
8. *Select values 2*
9. *Sort rows (Final2_IN)*
10. *Filter (Not null)*
11. *Filter week* (período de análise)
12. *File Output (.csv)*

Figura 2 – Etapas de pré-processamento e extração de dados do banco de dados *Vehicle Workplace* utilizando o software Pentaho.



Fonte: Elaborado pelo autor.

Etapas de pré-processamento e mineração do banco de dados *Status List (Alarmes)*:

1. *File input (.xlsx)*
2. *Select Values*
3. *Sort rows (linha)*
4. *Value mapper (Parada Linha)*
5. *Filter rows (Linha)*
6. *Sort rows (Duração)*
7. *Select values 2*
8. *StatusList_Output (.csv)*

Figura 3 – Etapas de pré-processamento e extração de dados do banco de dados *Status List (Alarmes)* utilizando o Pentaho.



Fonte: Elaborado pelo autor.

3 MATERIAIS E MÉTODOS

3.1 DESCRIÇÃO DA BASE DE DADOS

No desenvolvimento do algoritmo foram utilizadas as bases de dados de dados utilizadas *Vehicle Workplace* e *Status List (Alarmes)*, no período de fevereiro de 2023 a junho de 2023. Ver figuras 4 e 5.

Figura 4 – Informação banco de dados *Vehicle Workplace*.

Dataset statistics		Variable types	
Number of variables	25	Numeric	23
Number of observations	2924	DateTime	2
Missing cells	0		
Missing cells (%)	0.0%		
Duplicate rows	0		
Duplicate rows (%)	0.0%		
Total size in memory	593.9 KiB		
Average record size in memory	208.0 B		

Fonte: Elaborado pelo autor.

Figura 5 – Informação banco de dados *Status List* (Alarmes)

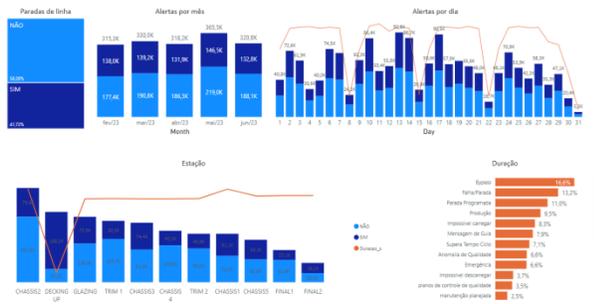
Dataset statistics		Variable types	
Number of variables	5	Categorical	3
Number of observations	1048576	DateTime	1
Missing cells	0	Numeric	1
Missing cells (%)	0.0%		
Duplicate rows	0		
Duplicate rows (%)	0.0%		
Total size in memory	40.0 MiB		
Average record size in memory	40.0 B		

Fonte: Elaborado pelo autor.

3.2 ANÁLISE DESCRITIVA DOS DADOS

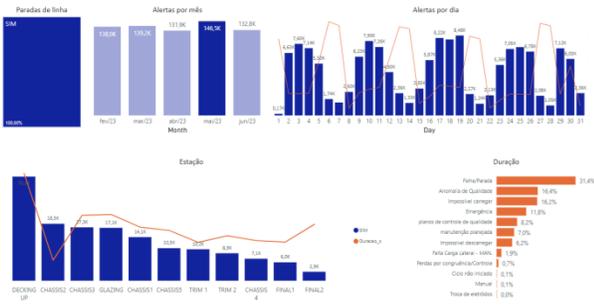
Foi realizada a análise do banco de dados *Status List* (Alarmes), verificar figuras 8 e 9.

Figura 6 – Dashboard para análise descritiva e exploratória.



Fonte: Elaborado pelo autor.

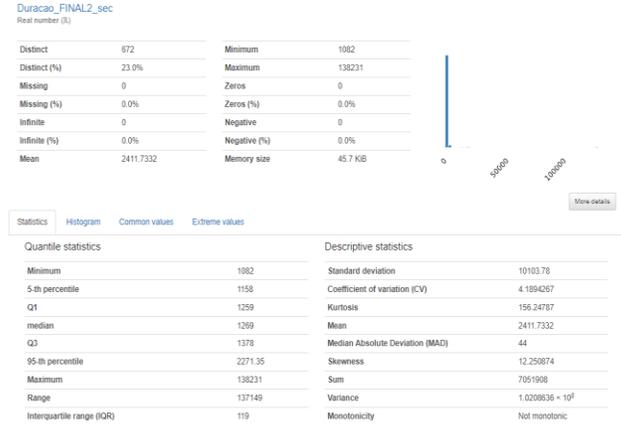
Figura 7 – Dashboard com filtro de alarmes que geram paradas na linha produtiva aplicado.



Fonte: Elaborado pelo autor.

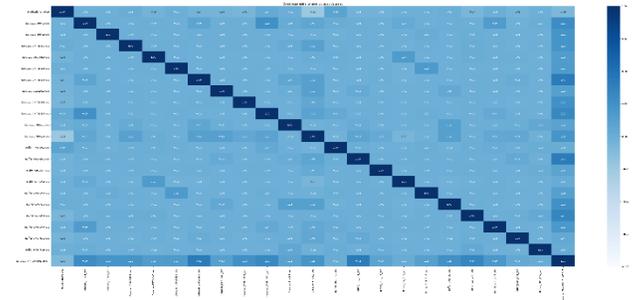
Para um melhor entendimento do comportamento dos dados foi elaborada a análise exploratória e descritiva dos dados, exemplificada nas figuras 6 e 7 para a base *Vehicle Workplace*.

Figura 8 – Exemplo análise exploratória dos bancos de dados.



Fonte: Elaborado pelo autor.

Figura 9 – Realizada correlação linear entre variáveis.

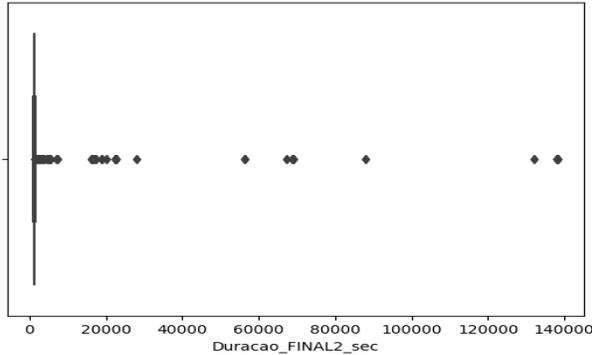


Fonte: Elaborado pelo autor.

3.3 PRÉ-PROCESSAMENTO DOS DADOS

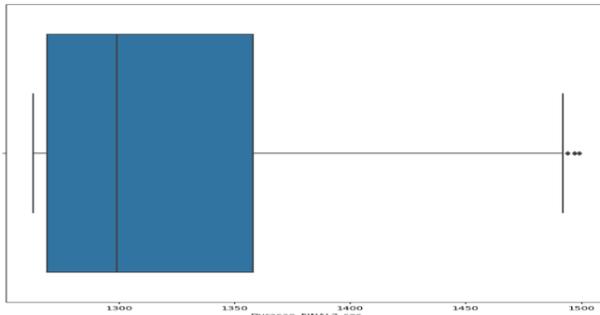
Foi estabelecida mais uma etapa de pré-processamento de dados para eliminar fontes de distorções que possam comprometer as etapas posteriores [5,6,7]. Ver figuras 10, 11 e 12.

Figura 10 – Elaboração do *Boxplot* da variável alvo (*Target*) antes do pré-processamento



Fonte: Elaborado pelo autor.

Figura 11 – Elaboração do *Boxplot* da variável alvo (*Target*) após o pré-processamento



Fonte: Elaborado pelo autor.

3.4 METODOLOGIA EXPERIMENTAL

O processo produtivo em estudo fornece uma enorme quantidade de informações sobre os tempos e alarmes do processo [4]. Ao enfrentar essa massa de dados precisamos criar um modelo de previsão eficaz e capaz de gerar informação de qualidade [8]. A técnica de seleção de características é uma solução que pode ser usada para selecionar variáveis que possuem maior impacto no resultado do modelo. Com base nesta premissa, é proposto um algoritmo *XGBoost* aprimorado baseado na seleção de importância de recursos [8]. O modelo foi treinado tomando como alvo o tempo de passagem no último posto de trabalho, utilizando o algoritmo *XGBoost* como regressor. Dessa forma o modelo deve compor os tempos de passagem em todos os postos de trabalho e prever o tempo de passagem no posto final. Ver figuras 12, 13, 14, e 15.

Figura 12 – Divisão do banco de dados e ajuste de parâmetros.

```
[344] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)

[345] params = {
    "objective": "reg:squarederror",
    "n_estimators": 300,
    "max_depth": 30,
    "learning_rate": 0.01,
    "subsample": 0.8,
    "colsample_bytree": 0.9,
    "colsample_bylevel": 0.8,
    "reg_lambda": 0.8,
    "eval_metric": "rmse",
    "random_state": 42,
}
```

Fonte: Elaborado pelo autor.

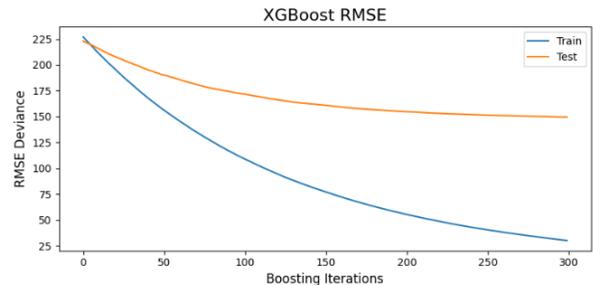
Figura 13 – Treinamento do modelo.

```
[346] reg = xgb.XGBRegressor(**params)

[347] reg.fit(X_train,
            y_train,
            verbose=False,
            eval_set= [(X_train, y_train), (X_test, y_test)],
            early_stopping_rounds= 3
            )
```

Fonte: Elaborado pelo autor.

Figura 14 – Elaboração do RMSE para avaliação do modelo.



Fonte: Elaborado pelo autor.

Figura 15 – Implementação do *feature Importance*, para extração de característica com maior impacto em relação ao tempo de fabricação do veículo.

```
] reg.feature_importances_

array([0.01807798, 0.03371179, 0.05164868, 0.03927552, 0.02526671,
       0.03311942, 0.02630008, 0.08266865, 0.03497706, 0.04936337,
       0.04474709, 0.05676966, 0.05239682, 0.03972628, 0.05388261,
       0.03053596, 0.15780668, 0.03664792, 0.04118643, 0.09277117],
      dtype=float32)

feat_importance = ["weight", "gain", "cover", "total_gain", "total_cover"]

for i in feat_importance:
    importance = reg.get_booster().get_score(importance_type=i)
    sorted_importance = sorted(importance.items(), key=lambda x: x[1], reverse=False)
    features, scores = zip(*sorted_importance)

fig, ax = plt.subplots(figsize=(12, 6)) # Ajuste o tamanho do gráfico conforme necessário
ax.barh(range(len(scores)), scores, tick_label=features)

ax.set_ylabel(i)
ax.set_title(f"Feature Importance - {i}") # Definir o título do gráfico

plt.show()
```

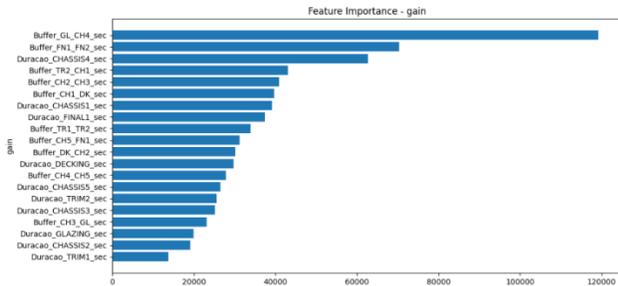
Fonte: Elaborado pelo autor.

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

4.1 RESULTADOS

A partir do modelo treinado foi feita a extração de características mais significantes para o resultado do modelo. Ver Figura 16.

Figura 16 – Gráfico com características mais significantes ordenadas da maior para menor.

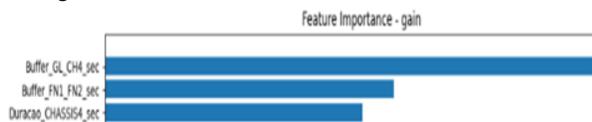


Fonte: Elaborado pelo autor.

4.2 DISCUSSÃO

A partir dos resultados estabelecidos é possível fazer um cruzamento de dados entre os dois bancos de dados que trabalhamos no projeto, chegando a uma informação de valor na qual podemos identificar o posto de trabalho que tem maior influência no resultado do modelo e nesse posto de trabalho quais os principais problemas que causam perdas no processo produtivo. Ver figuras 17 e 18.

Figura 17 – Recorte do gráfico com as três características mais significantes.



Fonte: Elaborado pelo autor.

Figura 18 – Gráfico com aplicação de filtro da principal característica identificada na Figura 17.



Fonte: Elaborado pelo autor.

Dessa forma, conseguimos identificar o posto de maior impacto e as perdas associadas a ele.

5 CONCLUSÕES E TRABALHOS FUTUROS

Na conclusão deste estudo, destacamos os resultados promissores alcançados ao empregar o algoritmo de *machine learning* na criação de um modelo de regressão para analisar e identificar as principais características impactantes nas perdas do processo em questão.

A capacidade do modelo em destacar as *features* de maior influência oferece não apenas uma compreensão aprofundada dos mecanismos subjacentes ao processo, mas também abre caminho para estratégias de otimização e mitigação mais direcionadas, reduzindo os custos produtivos. A interpretabilidade do XGBoost se mostrou crucial, fornecendo uma visão clara sobre como cada variável contribui para as perdas, permitindo uma tomada de decisão mais assertiva.

Os resultados validam a utilidade do XGBoost como uma ferramenta robusta para modelagem de regressão em contextos complexos, onde a identificação de padrões sutis é essencial. No entanto, é importante destacar que, como em qualquer abordagem analítica, a interpretação dos resultados deve ser realizada considerando o contexto específico do problema e suas nuances.

Como trabalho futuro, sugerimos a continuidade da pesquisa, explorando a aplicação do modelo em situações dinâmicas e avaliando sua robustez em diferentes cenários. Além disso, a incorporação de dados adicionais pode enriquecer ainda mais a capacidade preditiva do modelo, aprimorando sua generalização para casos mais amplos.

Em última análise, este estudo contribui não apenas para a compreensão das causas das perdas no processo, mas também para o avanço no uso de técnicas de aprendizado de máquina na otimização de processos industriais.

REFERÊNCIAS

- [1] L. C. Corrales, M. P. Lambán, M. E. H. Korner, J. Royo, **Overall Equipment Effectiveness: Systematic Literature Review and Overview of Different Approaches**, Applied Sciences 10 (2020) p. 6469.
- [2] Dobra, Péter; Jósваи, János. **Overall Equipment Effectiveness (OEE) Life Cycle at the Automotive Semi-Automatic Assembly Lines**, Acta Polytechnica Hungarica, Vol. 19, No. 9, 2022.

- [3] D. Mourtzis, E. Vlachou, N. Milas, **Industrial Big Data as a Result of IoT Adoption in Manufacturing**, *Procedia CIRP* 55 (2016), pp. 290-295.
- [4] DUMBILL, E. . **What is big data?**. O'Reilly Radar, 2012. Disponível em :<<http://radar.oreilly.com/2012/01/what-is-big-data.html>>. Acesso em: 23 nov. 2023.
- [5] B. K.,. S. G. M. ' F. H. Sergio Ramírez-Gallego, **A survey on data preprocessing for data stream mining: Current status and future directions**, Elsevier, pp. 39-57, 2017.
- [6] Viana, E., Silva, G., Guerra, L., & da Paz, W. (2021). **Previsão de Duração de Paradas de Linha em uma Indústria Automotiva**. *Revista De Engenharia E Pesquisa Aplicada*, 6(3), 49-58.
- [7] **Rodriguez Rodriguez, Oscar Wilfredo**. Pré-processamento de dados na identificação de processos industriais. **Dissertação de mestrado, Escola politécnica de São Paulo-SP, 2015**.
- [8] Ben Jabeur, S., Stef, N. & Carmona, P. **Bankruptcy Prediction using the XGBoost Algorithm and Variable Importance Feature Engineering**. *Comput Econ* 61, 715–741 (2023).