

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

EDSON ARTHUR ZANCHETA

**OTIMIZAÇÃO DE PROCESSOS EM UMA MONTADORA
AUTOMOTIVA: EFICÁCIA GERAL DO EQUIPAMENTO (OEE)
ATRAVÉS DA ANÁLISE DE DADOS COM PYTHON**

SÃO CARLOS
2024

EDSON ARTHUR ZANCHETA

**OTIMIZAÇÃO DE PROCESSOS EM UMA MONTADORA
AUTOMOTIVA: EFICÁCIA GERAL DO EQUIPAMENTO (OEE)
ATRAVÉS DA ANÁLISE DE DADOS COM PYTHON**

Trabalho de Conclusão de Curso
apresentado ao Departamento
de Engenharia Mecânica da
Universidade Federal de São Carlos,
para obtenção do título de Bacharel
em Engenharia Mecânica.

Orientador: Prof. Dr. João Vitor
de Carvalho Fontes

SÃO CARLOS
2024



FUNDAÇÃO UNIVERSIDADE FEDERAL DE SÃO CARLOS

COORDENAÇÃO DO CURSO DE ENGENHARIA MECÂNICA - CCEMec/CCET

Rod. Washington Luís km 235 - SP-310, s/n - Bairro Monjolinho, São Carlos/SP, CEP 13565-905

Telefone: (16) 33519703 - <http://www.ufscar.br>

DP-TCC-FA nº 4/2024/CCEMec/CCET

Graduação: Defesa Pública de Trabalho de Conclusão de Curso

Folha Aprovação (GDP-TCC-FA)

FOLHA DE APROVAÇÃO

EDSON ARTHUR ZANCHETA

OTIMIZAÇÃO DE PROCESSOS EM UMA MONTADORA AUTOMOTIVA: EFICÁCIA GERAL DO EQUIPAMENTO (OEE) ATRAVÉS DA ANÁLISE DE DADOS COM PYTHON

Trabalho de Conclusão de Curso

Universidade Federal de São Carlos – Campus São Carlos

São Carlos, 02 de fevereiro de 2024

ASSINATURAS E CIÊNCIAS

Cargo/Função	Nome Completo
Orientador	João Vitor de Carvalho Fontes
Membro da Banca 1	Gustavo Franco Barbosa
Membro da Banca 2	Joao Gustavo Pereira da Silva



Documento assinado eletronicamente por **Joao Vitor de Carvalho Fontes, Docente**, em 02/02/2024, às 11:25, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Joao Gustavo Pereira da Silva, Docente**, em 02/02/2024, às 11:27, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Gustavo Franco Barbosa, Professor(a) do Ensino Superior**, em 02/02/2024, às 11:27, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site <https://sei.ufscar.br/autenticacao>, informando o código verificador **1342674** e o código CRC **80EB7773**.

Referência: Caso responda a este documento, indicar expressamente o Processo nº 23112.002311/2024-27

SEI nº 1342674

Modelo de Documento: Grad: Defesa TCC: Folha Aprovação, versão de 02/Agosto/2019

Criado por 709935, versão 2 por 709935 em 30/01/2024 11:35:42.

AGRADECIMENTOS

Gostaria de agradecer as minhas irmãs Raissa, Carina e Sophia, por toda a participação na formação do meu caráter e da minha personalidade. Agradeço aos meus pais por todo o apoio e paciência, foi um caminho complexo, de altos e baixos e recebi muito mais do que merecia, obrigado. Agradeço ao basquete e ao vôlei UFSCar pela serenidade e refúgio que precisei durante toda a graduação, fiz meus amigos e conheci pessoas incríveis. Agradeço em especial ao professor João Vitor de Carvalho Fontes, que foi paciente e calmo durante todo o estudo, me ajudou durante a minha formação e trouxe o último fôlego que eu precisava na formação. Aos meus amigos e minha casa em São Carlos, a república Boate Azul, que me acolheu e me deu amigos que pude compartilhar a rotina e me fizeram ser uma pessoa melhor. Agradeço também à pessoas importantes para mim, que estiveram do meu lado, como Vitória, Derick, Diego, Mateus, Rafaela, Igor, Galhardo, Marcelo, Vitor, João, Jadiel, André, Osmar, Elisa e Daiane. Obrigado a todos por participarem por tanto tempo e com tanto afeto.

Lamentar profundamente é ter amado plenamente.

RESUMO

A indústria evoluiu com novas abordagens de trabalho e ampliou a relevância da integração de análises avançadas de dados e técnicas de aprendizado de máquina na engenharia, em busca por preços competitivos através da eliminação de desperdícios para aumentar os lucros. Dessa forma, este trabalho aborda a integração da Eficácia Geral do Equipamento (OEE) em um processo de estampagem em uma montadora automotiva por meio da análise de dados com Python. O OEE é uma métrica crucial na engenharia industrial responsável por medir a eficácia geral usando três componentes: disponibilidade, desempenho e qualidade. Já o Python é uma ferramenta com recursos abrangentes de processamento de dados e aprendizado de máquina. A metodologia adotada envolve o pré-processamento de dados, cálculo dos componentes do OEE e a aplicação de técnicas de aprendizado de máquina para prever e aprimorar eficiências operacionais. Este processo visa o cálculo do OEE e estabelecer a base para estratégias de manutenção preditiva e otimização de processos. O uso de aprendizado de máquina permite analisar e interpretar dados de máquinas e processos, possibilitando previsões de falhas, otimizações de cronogramas de manutenção e melhorias na eficiência operacional. Os resultados expõem quais peças demonstram maior eficiência na sua produção, o OEE da produção geral e um exemplo de predição de 10 lotes de peças. Dessa maneira, o estudo contribui para o potencial transformador da ciências de dados na engenharia mecânica, estabelecendo um precedente para futuras pesquisas e aplicações na indústria.

Palavras-chave: Eficiência Geral dos Equipamentos, Python, Engenharia Mecânica, Indústria Automotiva, Análise de Dados, Aprendizado de Máquina, Otimização de Processos.

ABSTRACT

The industry has evolved with new work approaches and increased the relevance of integrating advanced data analytics and machine learning techniques in engineering, in search of competitive prices by eliminating waste to increase profits. Therefore, this work addresses the integration of Overall Equipment Effectiveness (OEE) in a stamping process in an automotive assembly plant through data analysis with Python. OEE is a crucial metric in industrial engineering responsible for measuring overall effectiveness using three components: availability, performance and quality. Python is a tool with comprehensive data processing and machine learning capabilities. The methodology adopted involves pre-processing data, calculating OEE components and applying machine learning techniques to predict and improve operational efficiencies. This process aims to calculate OEE and establish the basis for predictive maintenance and process optimization strategies. The use of machine learning allows you to analyze and interpret data from machines and processes, enabling failure predictions, optimizations of maintenance schedules and improvements in operational efficiency. The results show which parts demonstrate greater efficiency in their production, the OEE of general production and an example of prediction of 10 batches of parts. In this way, the study contributes to the transformative potential of data sciences in mechanical engineering setting a precedent for future research and applications in industry.

Keywords: Overall Equipment Effectiveness, Python, Mechanical Engineering, Automotive Industry, Data Analysis, Machine Learning, Process Optimization.

LISTA DE FIGURAS

Figura 1 – Uma estamparia padrão em um montadora automotiva	20
Figura 2 – Árvore de Processos de Prensagem	21
Figura 3 – Modelo de perdas e processo para o OEE	23
Figura 4 – Benefícios e limitações do OEE	24
Figura 5 – Estrutura das 6 perdas de tempo observadas no indicador de OEE	27
Figura 6 – Tipos de <i>Machine Learning</i>	31
Figura 7 – Tarefas realizadas para o aprendizado de máquinas	32
Figura 8 – Ecossistema do <i>Machine Learning</i>	33
Figura 9 – Passo a passo na estampagem de uma peça em uma montadora automotiva	36
Figura 10 – Média do Tempo Total de Produção por Tipo de Peça.	45
Figura 11 – Relação entre ADC e Tempo Total de Produção	46
Figura 12 – Média de ADC por Tipo de Peça.	46
Figura 13 – Média geral do OEE.	47
Figura 14 – OEE individuais - Grupo 1	48
Figura 15 – OEE individuais - Grupo 2	48
Figura 16 – OEE individuais - Grupo 3	48
Figura 17 – OEE individuais - Grupo 4	48
Figura 18 – OEE individuais - Grupo 5	49
Figura 19 – OEE individuais - Grupo 6	49
Figura 20 – OEE individuais - Grupo 7	49
Figura 21 – OEE individuais - Grupo 8	49
Figura 22 – Comparação da predição e do resultado real dos OEE	52

LISTA DE TABELAS

Tabela 1 – Estrutura e descrição do <i>DataFrame</i>	41
Tabela 2 – Dados utilizados para a previsão do OEE	50

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application programming interface</i>
ADC	<i>Auto Die Change</i>
CLP	Controlador Lógico Programável
D	Disponibilidade
Df	<i>Dataframe</i>
DB	<i>Data Block</i>
IA	Inteligencia Artificial
ML	<i>Machine Learning</i>
MSE	<i>Mean Squared Error</i>
OEE	<i>Overall Equipment Effectiveness</i>
P	Performance
Q	Qualidade
SPM	<i>Strokes per minute</i>
Ton	Toneladas
TPM	<i>Total Productive Maintenance</i>
TCC	Trabalho de Conclusão de Curso

LISTA DE SÍMBOLOS

μ	Média da distribuição
σ	Desvio padrão
σ^2	Variância

SUMÁRIO

1	INTRODUÇÃO	15
1.1	OBJETIVOS	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	PROCESSO DE ESTAMPAGEM	18
2.2	O IMPACTO DO <i>AUTO DIE CHANGE</i> (ADC)	20
2.3	EFICÁCIA GERAL DO EQUIPAMENTO (OEE)	22
2.3.1	Disponibilidade	24
2.3.2	Desempenho	25
2.3.3	Qualidade	26
2.3.4	Síntese e Classificação atual	26
2.4	ANÁLISE DE DADOS E CÁLCULO DE OEE	28
2.4.1	Aquisição e Limpeza de Dados	29
2.4.2	Visualização de Dados:	30
2.5	APRENDIZADO DE MÁQUINA PARA APRIMORAMENTO DE OEE	30
3	METODOLOGIA	35
3.1	ÁRVORE DE PROCESSOS DA ESTAMPAGEM	35
3.2	PRÉ-PROCESSAMENTO DE DADOS	38
3.3	GRANULAÇÃO E TRANSFORMAÇÃO	38
3.4	CÁLCULO DO <i>AUTO DIE CHANGE</i> (ADC)	39
3.5	CÁLCULO DO OEE	39
3.6	APRENDIZADO DE MÁQUINA	40
3.7	ESTRUTURA DO <i>DATAFRAME</i> E EXEMPLO DE DADOS	40
4	RESULTADOS	43
4.1	IMPORTAÇÃO E TRATAMENTO DOS DADOS	43
4.2	PIVOTAMENTO E PREPARAÇÃO	44
4.3	CÁLCULO DO ADC	44
4.4	CÁLCULO DO OEE	47
4.5	APRENDIZADO DE MÁQUINAS	50

5	CONCLUSÃO	53
	REFERÊNCIAS	55
	APÊNDICE A – IMPORTAÇÃO E TRATAMENTO INICIAL DOS DADOS	58
	APÊNDICE B – LIMPEZA E PIVOTAMENTO	59
	APÊNDICE C – CALCULO DO TEMPO DE TROCA DE MATRIZ	62
	APÊNDICE D – CALCULO DO OEE	65
	APÊNDICE E – APRENDIZADO DE MAQUINA SIMPLES	68

1 INTRODUÇÃO

As mudanças do mercado quanto a ciclos de vida mais curtos dos produtos e a incerteza na demanda afetam significativamente os sistemas de produção caracterizados por altos volumes e margens baixas. Um setor que utiliza sistemas de fabricação de linha de produção automatizados é o automotivo. Os desenvolvimentos tecnológicos estão remodelando os cenários tradicionais de manufatura, promovendo o surgimento de fábricas inteligentes ou da Indústria 4.0 (ÇiNAR et al., 2020). Nesse contexto, entender e melhorar a eficiência de equipamentos e processos é mais crítico do que nunca. Por exemplo, Hassani, Mazgualdi e Masrour (2019) mostram como o aprendizado de máquina pode prever e melhorar a eficiência do OEE em uma indústria. Tayal et al. (2020) introduziram uma abordagem para melhorar o OEE em pequenas indústrias, utilizando análises de Pareto e diagramas de Ishikawa.

Neste trabalho, estudaremos a fusão de dois componentes contemporâneos da manufatura: Eficiência Geral dos Equipamentos (OEE), como mostrado por (RABELO, 2016), e o uso de linguagens de programação. No caso desse estudo, será utilizado a linguagem Python, uma linguagem versátil e adaptável (SRINATH, 2017). O conceito base está na aplicação desses elementos em uma planta de estampagem de uma montadora automotiva, um componente crítico no processo de fabricação de veículos.

Além das peculiaridades da automação, o setor automotivo está focado na segurança, qualidade e sustentabilidade. Esses aspectos são fundamentais para a identidade da empresa no mercado (ZUCCOTTO, 2019). A eficácia dos sistemas de produção continua sendo o principal objetivo de cada indústria para ser competitiva e alcançar o sucesso, mas ainda é profundamente influenciada pelas demandas do mercado. Nesse contexto, a Manutenção Produtiva Total (TPM) é uma ferramenta industrial útil para melhorar a produtividade da planta e a eficiência operacional (ZENNARO et al., 2018).

A métrica central para medir o sucesso do programa de implementação do TPM é o índice OEE, combinando três dimensões de eficácia em um único valor:

disponibilidade, desempenho e qualidade. O OEE é a medida-chave para avaliar o desempenho de equipamentos individuais (NAKAJIMA, 1982).

Entendendo que o engenheiro não está mais limitado aos métodos tradicionais de projeto e análise, utilizaremos o Python como ferramenta de análise, contribuindo para processos mais robustos, confiáveis e eficientes. Python é uma linguagem de programação de alto nível e de propósito geral conhecida por sua simplicidade e eficiência no processamento de dados e nas bibliotecas de aprendizado de máquina. A integração do Python no âmbito da engenharia, particularmente nos processos de uma linha de montagem automotiva, é uma ferramenta de grande potencial para aprimorar a eficiência produtiva e o controle de qualidade (HANSEN, 2001). A integração da engenharia com análise de dados e aprendizado de máquina é um paradigma atraente que oferece potencial transformador para vários setores (YAZDI; AZIZI; HASHEMIPOUR, 2018).

O setor no qual aplicaremos o estudo é a estamparia de uma montadora automotiva, a estampagem é um componente crítico da fabricação de automóveis, onde a chapa metálica é moldada em peças específicas por meio de prensagem ou puncionamento. Dada a importância e complexidade deste processo, é importante assegurar o seu desempenho, onde entra o conceito de Eficácia Global do Equipamento (OEE) (YU, 2023).

Em essência, este estudo não é apenas uma exploração das bases teóricas do OEE, mas uma aplicação prática do Python em um cenário do mundo real. Ele exemplifica a convergência dos princípios da engenharia mecânica com a análise de dados, impulsionando esta integração para facilitar e otimizar métodos aplicados na indústria. Ao aproveitar a polivalência do Python na análise do OEE, este estudo contribui para a verificação da eficiência, confiabilidade e produtividade em processos mecânicos tradicionais dentro da indústria automotiva (SRINATH, 2017).

Portanto, o OEE e seus componentes são indicadores no arsenal de uma montadora automotiva. Eles oferecem uma visão abrangente do processo de produção, garantindo que os recursos sejam efetivamente utilizados e os produtos sejam de alta qualidade. Ao focar nessas métricas, as montadoras automotivas podem otimizar suas operações, resultando em maior eficiência, redução de custos e aumento da lucratividade (TSAROUHAS, 2020).

1.1 OBJETIVOS

O objetivo principal deste estudo é trazer a integração de um processo de produção tradicional, como a estampagem, integrado a análises de dados e técnicas de aprendizado de máquina. Esta integração visa melhorar as capacidades de tomada de decisão e a eficiência operacional, permitindo assim estratégias mais eficazes tanto na manutenção preditiva, como na otimização de processos.

Para atingir tal objetivo, os objetivos específicos são listados abaixo:

- Utilizar a linguagem Python para a importação, pré-processamento e detalhamento dos dados, assim calculando o OEE.
- Análise da situação e da eficiência da produção, trazendo métricas para observar a produção como um todo e depois avaliando o desempenho por cada tipo de peça produzida.
- Aplicação de aprendizado de máquina com a utilização de algoritmos simples para uma predição utilizando 10 lotes de peças aleatórios possíveis.

2 FUNDAMENTAÇÃO TEÓRICA

A eficiência na produção industrial é um tema crucial para o desenvolvimento e competitividade das indústrias. Diversas pesquisas abordam estratégias para melhorar esta eficiência, focando em tecnologias e métodos inovadores. A automação da produção é um fator chave para aumentar a eficiência industrial. Terehov, Zaikin e Averchenkov (2021) desenvolveram um módulo de software para automação do planejamento da produção, resultando em um aumento significativo da taxa de produção nas empresas.

Visando economia de recursos, Shinkevich et al. (2019) analisaram as principais tendências do conceito de “Indústria 4.0” na indústria petroquímica, destacando a automação dos processos de produção como um meio de melhorar a eficiência.

A otimização na estampagem pode ser analisada através do conceito do OEE. Yogesh e Prabakaran (2017) investigaram a implementação de uma metodologia focada em eliminar práticas desnecessárias em setores de pequenas e médias empresas, incluindo estampagem onde houve um aumento notável na eficiência na seção de montagem.

Nunes, Campilho e Silva (2022) analisaram e melhoraram a eficiência do processo de estampagem em um fabricante de componentes automotivos, cuja razão do baixo OEE estava na demora para trocar as matrizes. Propondo um novo design para o sistema de transferência, resultando em uma redução significativa dos tempos de configuração e paradas de produção, levando a uma maior confiabilidade.

2.1 PROCESSO DE ESTAMPAGEM

A linha de produção de estampagem, do ponto de vista da composição de equipamentos, geralmente é composta por um sistema de desempilhamento, um

sistema de alimentação, um sistema de descarga e um sistema de prensa. Com o rápido desenvolvimento da indústria automobilística, as linhas de produção de estampagem tradicionais, como a linha de produção de estampagem manual e a linha de produção de estampagem com dispositivos mecânicos simples, foram sendo substituídas por linhas de produção automáticas de estampagem para atender às necessidades de desenvolvimento em larga escala e alta qualidade na indústria. Atualmente, as linhas de produção automáticas de estampagem comuns incluem linhas de produção de estampagem de matriz progressiva, linhas de produção de estampagem multiestação e linhas de produção de estampagem em série (GEN; YUNONG, 2020).

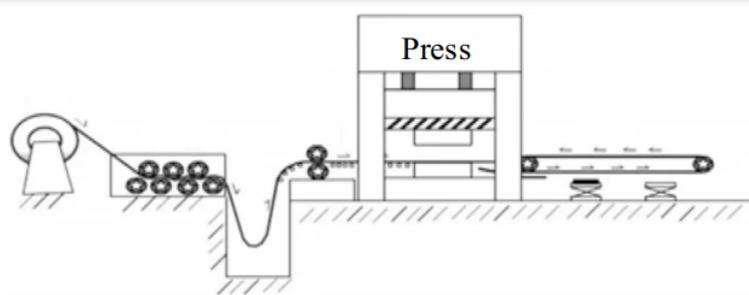
Na Figura 1a, a linha de produção automática de estampagem é uma linha de produção que utiliza uma matriz progressiva para estampar, geralmente composta pelo desenrolamento do material, alimentador, prensa e matriz. Ela pode realizar o processo automático de desenrolar a bobina, alargar e lubrificar a tira, estampar a chapa metálica em peças e coletar as peças finalizadas no final da linha (YU, 2023).

Na linha de produção automática de estampagem multiestação na Figura 1b, várias matrizes independentes estão colocadas na bancada da prensa com alta tonelagem. Desempilhadores ou alimentadores de desenrolamento, chamados de *destackers*, são usados para carregar os materiais. Garras robóticas de alimentação transferem as peças de trabalho. Esteiras transportadoras automáticas são introduzidas para coletar produtos fora da especificação planejada.

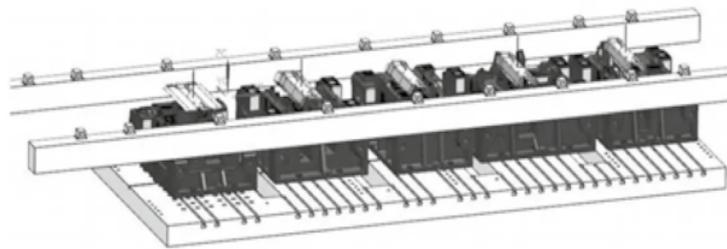
Na Figura 1c, a linha de produção automática de estampagem em série consiste em mecanismos de alimentação e prensas em série. Um conjunto de matrizes é colocado na bancada de prensa, onde manipuladores automáticos e robôs, são usados para pegar, transferir e posicionar as peças de estampagem.

As fábricas de montagem automotiva têm árvores de processo projetadas para garantir que cada peça seja trabalhada com precisão e com o mais alto padrão de qualidade. A oficina de estampagem assume um papel particularmente importante nesta cadeia, responsável pela criação das diversas peças metálicas que constituem a estrutura e os painéis do veículo (YU, 2023).

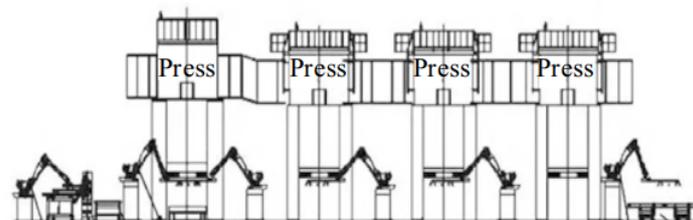
Figura 1 – Uma estamparia padrão em um montadora automotiva



(a) Progressive die stamping production line



(b) Multi-station stamping automatic production line



(c) Series stamping automatic production line

Fonte: Yu (2023)

2.2 O IMPACTO DO *AUTO DIE CHANGE* (ADC)

No processo de estampagem automotiva, a troca de moldes ou matrizes é uma operação vital que demanda extrema precisão e coordenação. Essa transição, do desmonte de uma matriz até a instalação e calibração da próxima, é fundamental para a eficiência e qualidade da produção. A continuidade e a fluidez do processo são cruciais para evitar paradas desnecessárias e garantir a uniformidade das peças

estampadas. Esta etapa também envolve a verificação da compatibilidade dos moldes com o equipamento, ajustes mecânicos e calibrações específicas para cada peça produzida. As técnicas de troca rápida de moldes são aplicadas para minimizar o tempo de inatividade. Na Figura 2 é detalhado a sequência de etapas no processo de troca de moldes, destacando a importância de cada fase e como ela se relaciona com os objetivos de eficiência e qualidade na produção de peças estampadas na indústria automotiva.

Figura 2 – Árvore de Processos de Prensagem



Fonte: Autor

2.3 EFICÁCIA GERAL DO EQUIPAMENTO (OEE)

A eficácia dos equipamentos desempenha um papel fundamental na indústria 4.0. Segundo Yazdi, Azizi e Hashemipour (2018) no ambiente dinâmico da manufatura, equipamentos de fabricação confiáveis e o OEE são os principais componentes para melhorar o desempenho e a lucratividade dos sistemas de fabricação.

O OEE emerge como uma métrica fundamental na engenharia industrial, oferecendo uma medida quantificável da eficiência do equipamento que agrega três dimensões centrais: disponibilidade, desempenho e qualidade. Seu cálculo proporciona aos gestores a possibilidade de identificar as principais perdas que reduzem a eficácia da máquina e então tomar as decisões necessárias para melhorar a situação (NAKAJIMA, 1982).

Introduzido por Nakajima (1982), o OEE era um dos pilares no conceito de Manutenção Produtiva Total (TPM) que surgiu no final da década de 1960 (NAKAJIMA, 1998). Desde então, tem sido amplamente adotado como um indicador-chave de desempenho nas indústrias, sendo um indicador da lacuna entre o desempenho real do processo de fabricação e o ideal. O conceito de *Overall Equipment Effectiveness* fornece uma medida abrangente dessa eficiência, medindo a eficiência de qualquer fábrica através da sua disponibilidade para produção, desempenho em termos de SPM (do inglês *strokes per minute*) e qualidade dos produtos que produz (SANTOS, 2018).

Nakajima (1998) sugere que para aprimorar o desempenho, é vantajoso utilizar a análise do indicador OEE juntamente com ferramentas de gestão da qualidade, tais como o gráfico de Pareto e o diagrama de causa e efeito. O gráfico de Pareto é eficaz na identificação das principais fontes de perda que mais afetam o OEE, enquanto o diagrama de causa e efeito auxilia na estruturação da investigação para descobrir as verdadeiras causas dessas perdas, tornando mais fácil sua identificação e resolução.

O modo de cálculo do OEE é sugerido por Nakajima (1982) e Groote (1995), com a seguinte fórmula 1:

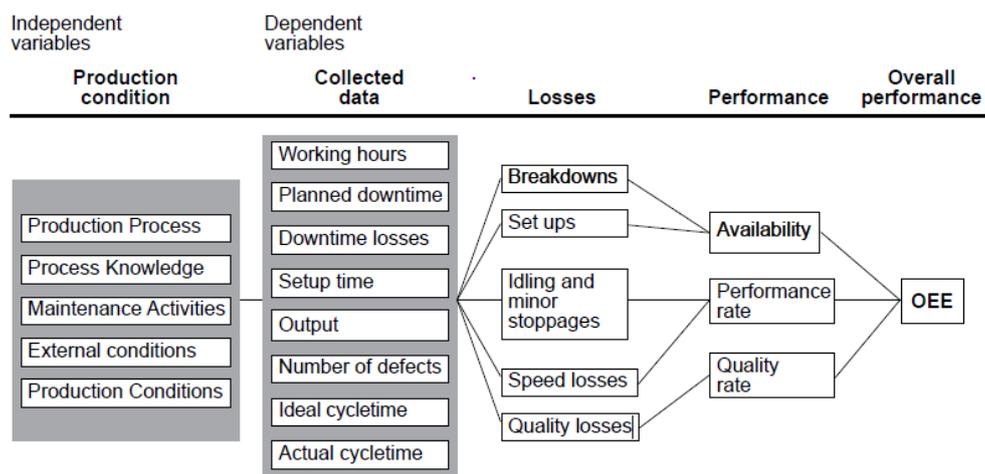
$$\text{OEE} = \text{Disponibilidade} \times \text{Desempenho} \times \text{Qualidade} \quad (1)$$

Para a verificação dos principais impactos em um OEE, Ljungberg (1998) desenvolveu um formulário para a coleta de dados. Neste estudo, formulários foram criados com colaboração direta de operadores e técnicos, coletando dados de perturbações durante quatro semanas. As perdas de produção foram registradas e analisadas, abrangendo aspectos como tempo de ciclo, defeitos de qualidade e manutenção, sendo correlacionadas com variáveis independentes em cinco categorias sendo elas

1. Processo de produção.
2. Conhecimento do processo.
3. Atividades de manutenção.
4. Condições externas.
5. Condições de produção

Essa análise detalhada contribuiu para uma compreensão mais profunda das causas das perdas na produção, o que é exemplificado na Figura 3.

Figura 3 – Modelo de perdas e processo para o OEE



Fonte: Ljungberg (1998)

O OEE não é um indicador completo assim como resalta Rabelo (2016). Uma grande fragilidade do OEE é a sua falta de visão da eficácia externa que o impede de refletir o que ocorre na cadeia de valor. Consistindo como resultado a dificuldade de percepção de *trade-offs* entre o objetivo da manufatura e o de outras áreas que interagem com a mesma (RABELO, 2016).

A figura 4 apresenta um resumo das principais vantagens e desvantagens associadas à utilização do OEE como um indicador de desempenho global na área de manufatura. Segundo Busso e Miyake (2013), é essencial procurar por métricas que não apenas mensurem o desempenho de equipamentos individuais ou áreas específicas, mas que também permitam a avaliação dos impactos em sistemas mais abrangentes dentro da cadeia de valor. Tais métricas devem ser aplicáveis tanto para comparações internas quanto externas entre organizações, devem levar em consideração as necessidades dos clientes e devem ser de fácil compreensão, atualização e implementação em processos de melhoria contínua (BUSSO; MIYAKE, 2013).

Figura 4 – Benefícios e limitações do OEE

Benefícios	Limitações
<ul style="list-style-type: none"> • Possibilita a análise de problemas de produção ou manutenção e consequente atuação na causa raiz (JEONG; PHILLIPS, 2001); • Possibilita a identificação de máquinas que devem ser foco de atividades de gestão da manutenção (BAMBER et al., 2003); • Permite comparação interna entre as máquinas de uma mesma planta (BAMBER et al., 2003); e • Registros de paradas para identificação das perdas permitem a complementação dos planos de manutenção já existentes (CHAND; SHIRVANI, 2000). 	<ul style="list-style-type: none"> • Quando aplicado a um escopo maior que uma única máquina (linha de produção ou planta), não direciona adequadamente as ações para melhoria contínua (BRAGLIA; FROSOLINI; ZAMMORI, 2009); • Não fornece visão sistêmica das perdas do negócio, pois não considera interações além do equipamento (JONSSON; LESSHAMMAR, 1999); • A utilização somente do OEE pode definir responsabilidades para a área de produção que não necessariamente são da mesma (LJUNGBERG, 1998); e • Dificuldade de reconhecer outras perdas com base na taxonomia das seis grandes perdas do OEE (JEONG; PHILLIPS, 2001).

Fonte: (BUSSO; MIYAKE, 2013)

2.3.1 Disponibilidade

A disponibilidade leva em consideração todos os eventos que interrompem a produção planejada por um período de tempo relevante (normalmente mais do que

alguns minutos). Isso inclui paradas não planejadas (como falhas de equipamento) e paradas planejadas (como trocas). É a medida de quantas vezes um sistema pode executar sua função projetada, quando necessário. No contexto da manufatura, a disponibilidade refere-se ao tempo real da máquina disponível para produção em comparação com o tempo de produção planejado. As paradas, são deduzidas do tempo total de produção planejado para calcular o tempo real da máquina como mostra a equação 2.

$$\text{Disponibilidade} = \frac{\text{Tempo de produção}}{\text{Tempo planejado de produção}} \quad (2)$$

Onde:

- **Tempo de Execução** = Tempo Planejado de Produção - Tempo de Parada
- **Tempo planejado de produção** = Tempo total programado para produção
- **Tempo de parada** = Duração total das paradas não planejadas

2.3.2 Desempenho

O desempenho considera os casos em que o processo é executado mais lentamente do que sua velocidade máxima teórica. Representa a relação de velocidade da produção real *versus* produção máxima. Os fatores que afetam o desempenho incluem pequenas paradas, marcha lenta e velocidades reduzidas. É calculado comparando o tempo de ciclo real (o tempo que uma máquina leva para completar um ciclo de operação) com o tempo de ciclo ideal (o tempo mais rápido possível para um ciclo) como mostra a equação 3.

$$\text{Desempenho} = \frac{\text{Total de peças produzidas}}{\text{Tempo Planejado de Produção} \times \text{Tempo Ideal de Ciclo}} \quad (3)$$

Onde:

- **Tempo de ciclo ideal** é o tempo de ciclo mais rápido para um produto (tempo necessário para uma máquina produzir uma peça em condições ideais).

2.3.3 Qualidade

A qualidade representa as unidades satisfatórias dentro das unidades produzidas. No contexto do OEE, denota a capacidade de produzir dentro do estipulado quando comparado com a produção total. Representa a parcela dos produtos dentro dos padrões preestabelecidos na quantidade total fabricada. A equação 4 abaixo mostra esse cálculo:

$$\text{Qualidade} = \frac{\text{Peças produzidas dentro do padrão de qualidade}}{\text{Total de Peças Produzidas}} \quad (4)$$

2.3.4 Síntese e Classificação atual

O cálculo do OEE é realizado mediante a identificação de seis categorias fundamentais de perdas, que são agrupadas em três classes distintas, como ilustrado na Figura 5 e também descrito por Vijayakumar e Gajendran (2014).

Onde cada categoria e classe estão descritas abaixo:

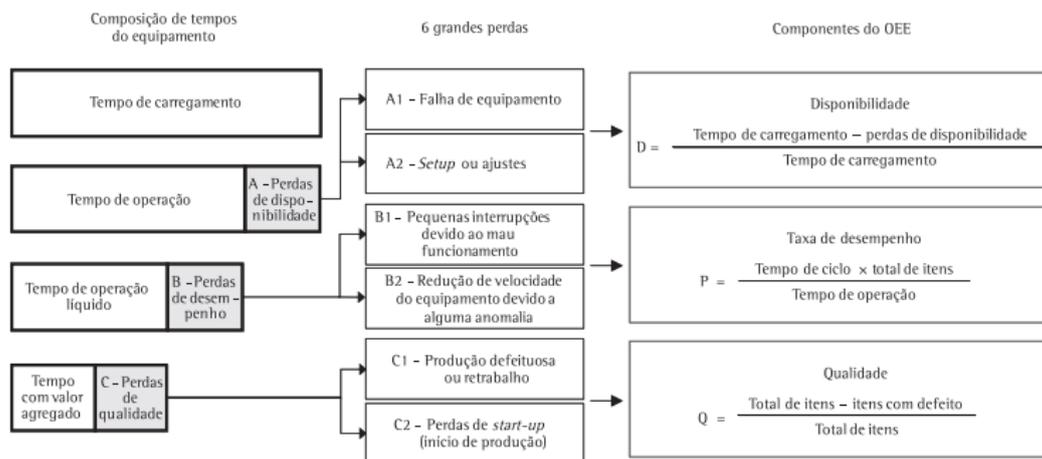
1. Perdas de Disponibilidade
 - A1. Paradas que causam falhas no funcionamento do equipamento.
 - A2. Paradas para a realização de configurações ou ajustes.
2. Perdas de Desempenho

- B1. Pequenas paradas ou interrupções devido a mau funcionamento do equipamento.
- B2. Redução da velocidade de operação do equipamento devido a alguma anomalia que resulta em um tempo de ciclo maior do que o tempo padrão.

3. Defeitos e Perdas de Qualidade

- C1. Produção defeituosa ou retrabalho.
- C2. Perdas durante a inicialização ou perdas ocorridas no início da produção devido a ajustes para estabilização do equipamento.

Figura 5 – Estrutura das 6 perdas de tempo observadas no indicador de OEE



Fonte: (BUSSO; MIYAKE, 2013)

De acordo com Hansen (2001), podemos qualificar o resultado calculado do OEE com a seguinte classificação:

- Menos de 65% - inaceitável, o processo requer ações imediatas;
- Entre 65% e 75% - o processo é considerado satisfatório;
- Entre 75% e 85% - o processo é classificado como muito bom, apresentando potencial para desempenho em nível global;

- Acima de 85% - o processo é considerado equivalente a uma empresa de classe mundial.

2.4 ANÁLISE DE DADOS E CÁLCULO DE OEE

Ao longo da história, uma variedade extensa de linguagens de programação e ambientes distintos foram empregados para auxiliar na pesquisa em aprendizado de máquina e no desenvolvimento de aplicações, uma das linguagens que emergiu notavelmente como a escolha dominante na comunidade de computação científica foi Python (SRINATH, 2017).

Caracterizada pela sua ênfase na legibilidade, o Python é uma linguagem de alto nível, conhecida por sua facilidade de aprendizado, enquanto mantém a capacidade de aproveitar as vantagens das linguagens de programação de baixo nível quando necessário. Além das características intrínsecas da linguagem, a presença de uma comunidade ativa e a ampla disponibilidade de ferramentas e bibliotecas fazem com que Python seja particularmente atraente para atividades relacionadas à ciência de dados, aprendizado de máquina e computação científica (RASCHKA; PATTERSON; NOLET, 2020).

Devido à sua natureza de tipagem dinâmica, o Python é altamente flexível. Além disso, o Python é mais tolerante a erros, permitindo que você compile e execute seu programa mesmo quando encontra partes problemáticas, processo chamado de “*debugging*”. É uma linguagem de programação versátil, capaz de suportar diferentes estilos de programação, incluindo abordagens estruturadas e orientadas a objetos (SAHOO et al., 2019). Além disso, é possível incorporar componentes modulares desenvolvidos em outras linguagens de programação no Python, o que significa que você pode escrever um programa em C++, por exemplo, e importá-lo como um módulo no Python, podendo ainda adicionar funcionalidades extras, como criar uma interface gráfica para ele (SRINATH, 2017).

Os métodos tradicionais de cálculo e avaliação do OEE podem ser com-

plicados e propensos a erros humanos. É aqui que Python, uma linguagem de programação de propósito geral e de alto nível, se mostra valiosa. A simplicidade e a eficiência do Python, juntamente com sua ampla seleção de processamento de dados e bibliotecas de aprendizado de máquina, o tornam uma ferramenta excelente. O Python pode lidar com eficiência com o processamento de dados necessário para calcular o OEE, permitindo assim avaliações mais precisas, consistentes e oportunas (KUMAR, 2015).

Para o aproveitamento total da ferramenta, não apenas foi realizado o cálculo do OEE, mas também as bases para a aplicação de técnicas de aprendizado de máquina. Segundo Maleki et al. (2020) os recursos de previsão e otimização oferecidos pelos algoritmos de aprendizado de máquina podem fornecer uma perspectiva adicional quanto ao processo. Com uma variedade de bibliotecas disponíveis, McKinney et al. (2011) ressalta o uso do Pandas, NumPy e Matplotlib e (PEDREGOSA et al., 2011) mostra as funcionalidades do pacote Sklearn, a combinação desses pacotes tem um grande potencial para a análise de dados (NONGTHOMBAM; SHARMA, 2021).

2.4.1 Aquisição e Limpeza de Dados

A aquisição e limpeza de dados com Python desempenham um papel vital na análise de dados e na tomada de decisões. A aquisição de dados envolve coletar informações de diversas fontes, como sensores em máquinas, bancos de dados e registros de sistemas de controle. Esses dados, muitas vezes brutos e desorganizados, requerem processamento para se tornarem úteis (SAHOO et al., 2019). A limpeza de dados é um processo crítico que envolve a remoção ou correção de dados imprecisos, duplicados ou mal formatados. Algumas ferramentas Python como Pandas e NumPy são essenciais para esta etapa, permitindo a manipulação eficiente de grandes conjuntos de dados e a aplicação de técnicas como a normalização, tratamento de valores ausentes e correção de erros (MCKINNEY et al., 2011)

Após a limpeza, os dados são geralmente explorados e analisados para

extrair *insights* valiosos. Neste estágio, visualizações de dados e algoritmos estatísticos são frequentemente utilizados para entender melhor as tendências e padrões nos dados. Esses processos são particularmente relevantes na manutenção preditiva, otimização de processos e controle de qualidade.

2.4.2 Visualização de Dados:

Com o uso do pacote Matplotlib, a visualização de dados auxilia na interpretação dos resultados, destacando tendências e *insights*. A visualização de dados com Python é uma ferramenta poderosa para interpretar e apresentar informações complexas de maneira acessível. Utilizando bibliotecas como Matplotlib, Seaborn e Plotly, pode-se criar gráficos, diagramas e mapas interativos que tornam os dados mais compreensíveis e acionáveis. Essas visualizações são cruciais em várias aplicações, como na manutenção preditiva, onde gráficos de tendências podem prever falhas em equipamentos, ou na otimização de processos, onde a visualização de dados de desempenho ajuda a identificar áreas de melhoria. Em resumo, a habilidade de visualizar dados com Python não apenas facilita a interpretação de dados complexos, mas também aprimora a tomada de decisões e a eficiência operacional (KUMAR, 2015).

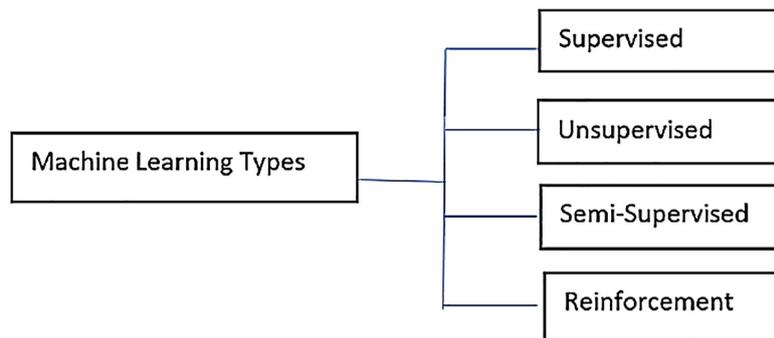
2.5 APRENDIZADO DE MÁQUINA PARA APRIMORAMENTO DE OEE

O aprimoramento do OEE através do aprendizado de máquina envolve o uso de algoritmos para analisar e interpretar dados de máquinas e processos. Com o aprendizado de máquina, é possível prever falhas, otimizar cronogramas de manutenção e melhorar a eficiência operacional (ÇINAR et al., 2020).

As técnicas de aprendizado de máquina são reconhecidas pela capacidade

de ensinar a máquina a realizar tarefas complexas, diferentemente dos métodos estatísticos simples, ao fornecer a essa máquina um conjunto de dados históricos. Dependendo do tipo de aprendizado de máquina utilizado, a máquina poderá prever novos dados de saída a partir dos dados de entrada fornecidos pelo usuário. Existem quatro tipos de aprendizado de máquina, como mostrado na Figura 6, e em nosso modelo, empregaremos o tipo de aprendizado de máquina supervisionado.

Figura 6 – Tipos de *Machine Learning*



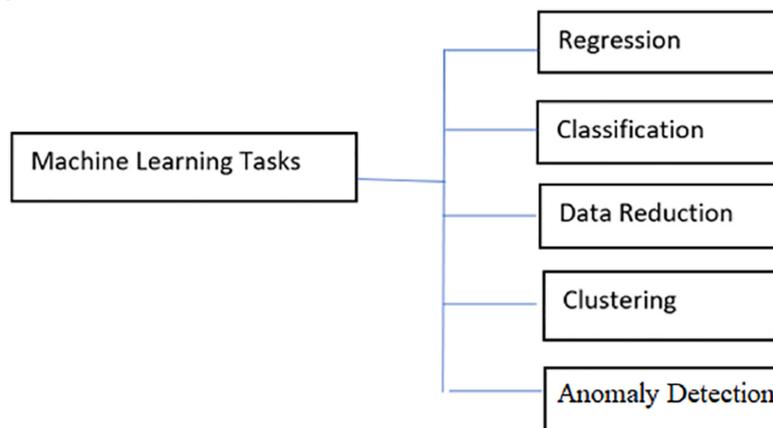
Fonte: (MJIMER; AOULA; ACHOUYAB, 2022)

Para o *Machine Learning* no estudo, foi utilizada a mesma abordagem utilizada por Mjimer, Aoula e Achouyab (2022), onde o aprendizado de máquina supervisionado realizará a tarefa de regressão. O uso do aprendizado supervisionado é indicado quando se tem conhecimento das características de entrada e há a necessidade de prever o valor de um ou mais resultados (HASTIE; TIBSHIRANI; FRIEDMAN, 2001).

Para as tarefas de aprendizado de máquina, a tarefa de regressão é reconhecida como um modelo preditivo que permite uma previsão para cada valor contínuo dos dados de saída (MALEKI et al., 2020). A Figura 7 ilustra os processos disponíveis.

A aplicação desta tecnologia permite identificar padrões ocultos nos dados operacionais e prever problemas antes que eles ocorram. Por exemplo, usando algoritmos de regressão e classificação, pode-se prever a probabilidade de falha de uma máquina, o que permite intervenções mais precisas e a redução do tempo de inatividade. Além disso, a análise de *clusters* pode ser usada para segmentar máquinas ou processos com base em seu desempenho, auxiliando na identificação

Figura 7 – Tarefas realizadas para o aprendizado de máquinas

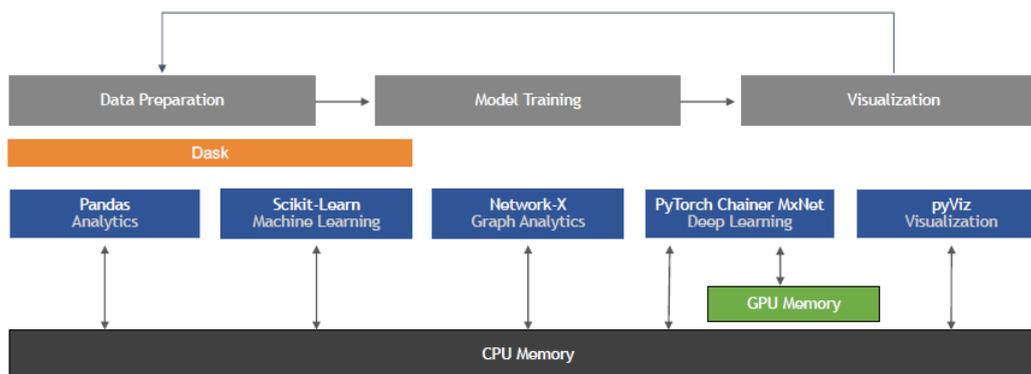


Fonte: (MJIMER; AOULA; ACHOUYAB, 2022)

de áreas que necessitam de melhorias (CADAVID et al., 2020).

Essas técnicas não apenas aumentam a eficiência do equipamento, mas também contribuem para a redução de custos operacionais, otimização do uso de recursos e melhoria contínua dos processos de produção. Com a integração do aprendizado de máquina, a gestão da OEE se torna mais dinâmica e adaptativa, levando a uma produção mais eficiente e sustentável (RABELO, 2016).

Enquanto tanto NumPy quanto Pandas (ver Figura 8) oferecem abstrações sobre um conjunto de pontos de dados com operações que funcionam no conjunto de dados como um todo, Pandas amplia o NumPy ao fornecer um objeto semelhante a um quadro de dados que suporta tipos de colunas heterogêneos e metadados de linhas e colunas. Nos últimos anos, a biblioteca Pandas tornou-se o formato padrão para representar dados tabulares em Python em contextos de extração, transformação e carga (ETL) e análise de dados (MCKINNEY et al., 2011). Doze anos após o seu primeiro lançamento em 2008 e 25 versões posteriores, a primeira versão 1.0 do Pandas foi lançada em 2020. Na comunidade de código aberto, onde a maioria dos projetos segue padrões de versionamento semântico, um lançamento 1.0 indica que uma biblioteca atingiu um alto nível de maturidade, juntamente com uma API estável (RASCHKA; PATTERSON; NOLET, 2020). Pandas, NumPy e SciPy continuam sendo as escolhas mais amigáveis e recomendadas para muitos projetos de ciência de dados e computação (KUMAR, 2015).

Figura 8 – Ecossistema do *Machine Learning*

Fonte: (RASCHKA; PATTERSON; NOLET, 2020)

O pacote *Scikit-learn*, como uso é ilustrado pela Figura 8 se tornou a biblioteca padrão da indústria em Python utilizada para engenharia em razão das características e modelagem de aprendizado de máquina clássica em conjuntos de dados pequenos a médios, em grande parte devido à sua API limpa, consistente e intuitiva (PEDREGOSA et al., 2011). Nesse contexto, como uma regra prática, considera-se conjuntos de dados com menos de 1000 exemplos de treinamento como pequenos e conjuntos de dados com entre 1000 e 100.000 exemplos como médios. Além disso, com a ajuda da comunidade de código aberto, a equipe de desenvolvedores do *Scikit-learn* mantém um forte foco na qualidade do código e na documentação abrangente. Utilizando o modelo de API simples, baseado no “*fit()/predict()*”, seu design tem servido como inspiração e modelo para muitas bibliotecas, pois apresenta uma interface familiar e reduz as mudanças de código quando os usuários exploram diferentes opções de modelagem (RASCHKA; PATTERSON; NOLET, 2020).

Um ponto da indústria 4.0 é a manutenção preditiva, sendo uma abordagem proativa para a manutenção de equipamentos e maquinários. Ela utiliza dados de monitoramento em tempo real e análises preditivas para prever falhas antes que ocorram. Isso envolve o uso de sensores, integração de tecnologias e técnicas avançadas de análise de dados, como aprendizado de máquina e inteligência artificial (NONGTHOMBAM; SHARMA, 2021). Ao identificar padrões e sinais de desgaste iminente, a manutenção preditiva permite intervenções oportunas, evitando falhas

inesperadas e reduzindo o tempo de inatividade. Essa abordagem melhora a eficiência operacional, prolonga a vida útil do equipamento e reduz custos relacionados a manutenções não planejadas (ÇINAR et al., 2020). Alguns pontos relevantes para a OEE e a manutenção preditiva são:

- **Previsão de Qualidade:** Algoritmos de classificação antecipam defeitos nos produtos, com base em parâmetros de produção.
- **Avaliação de Modelo:** O desempenho é avaliado usando precisão, acurácia, revocação e pontuação F1, com validação cruzada para verificação de robustez (MALEKI et al., 2020).
- **Melhoria Contínua:** Um ciclo de feedback garante que os modelos se adaptem às mudanças nas condições do processo de fabricação.

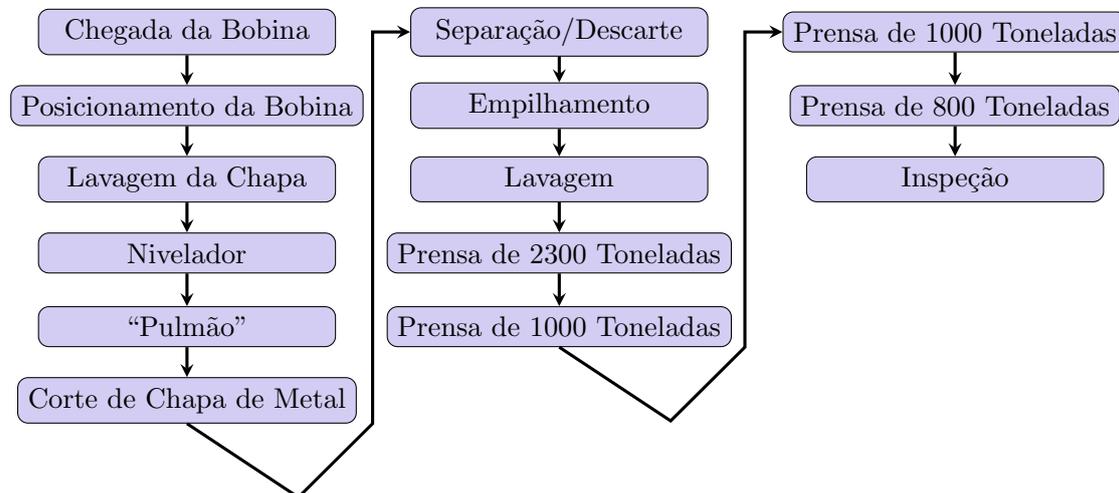
3 METODOLOGIA

A metodologia adotada aqui envolve o pré-processamento de dados, cálculo dos componentes do OEE e aproveita técnicas de aprendizado de máquina para prever e aprimorar eficiências operacionais (SAHOO et al., 2019). Esta metodologia explora a aplicação de Python na otimização do processo de estampagem em uma montadora automotiva, com ênfase no cálculo da Eficiência Geral do Equipamento (OEE) e a aplicação de algoritmos de aprendizado de máquina.

3.1 ÁRVORE DE PROCESSOS DA ESTAMPAGEM

Para o início do estudo, foi realizado o mapeamento da árvore de processos da estamparia presente na montadora, a Figura 9 mostra as etapas da chegada de uma bobina até a saída da peça estampada:

Figura 9 – Passo a passo na estampagem de uma peça em uma montadora automotiva



Fonte: Autor

Abaixo, temos o detalhamento e uma síntese de cada item da Figura 9:

- **Chegada da bobina:**

Este é o início do processo de estampagem, onde grandes bobinas de metal, principalmente aço ou alumínio, são enviadas para a instalação. Essas bobinas geralmente pesam várias toneladas e são transportadas em caminhões-plataforma ou ferrovias, dependendo da logística.

- **Nivelador:**

Após a lavagem, a folha de metal é alimentada em uma máquina de nivelamento. Esta máquina é composta por uma série de rolos que aplainam a chapa, garantindo que ela esteja livre de deformações ou imperfeições que possam ter ocorrido durante o enrolamento ou transporte.

- **“Pulmão”:**

O “pulmão” é essencialmente um *buffer* de armazenamento ou um sistema de *loop*. Ele compensa as diferenças na velocidade de processamento entre a máquina de nivelamento e as operações a frente, garantindo um fluxo de produção contínuo e suave.

- **Corte de Chapa Metálica:**

Uma vez que a folha é nivelada e protegida, ela é movida para a máquina de corte, que apara e corta a folha de metal no tamanho necessário para estampagem.

- **Separação/Eliminação de Defeitos:**

Após o corte, cada folha é inspecionada rapidamente. As folhas defeituosas ou com imperfeições visíveis são separadas e recicladas ou descartadas, dependendo da natureza e extensão do defeito.

- **Empilhamento:**

As chapas metálicas qualificadas são empilhadas em empilhadeiras automatizadas, preparando-as para os processos de estampagem subsequentes.

- **Lavar:**

Uma lavagem secundária é realizada para garantir que as folhas estejam perfeitamente limpas antes da estampagem, pois qualquer resíduo pode afetar a qualidade da estampagem.

- **Prensagem na prensa de 2300 toneladas:**

Primeiro, eles são alimentados em uma prensa de 2.300 toneladas, que pode ser responsável pela modelagem inicial.

- **Prensagem na prensa de 1000 toneladas:**

A folha então passa pela primeira prensa de 1000 Ton. Esta prensa pode executar operações de estampagem mais detalhadas ou intrincadas, refinando a forma da peça.

- **Prensagem na prensa de 1000 toneladas:**

Uma prensagem subsequente é feita por outra prensa de 1000 Ton, refinando ainda mais a forma da peça.

- **Prensagem na prensa de 800 toneladas:**

Por fim, a prensa de 800 Ton faz os retoques finais, dando à peça metálica sua forma e dimensões acabadas.

- **Inspeção:**

Após as operações de prensagem, cada peça estampada passa por uma inspeção. Quaisquer peças não conformes são enviadas de volta para retrabalho ou descartadas.

É importante o conhecimento do processo para lidar com os dados e os *outliers*, conseguindo interpretar com uma maior acurácia os tipos de percalços e gargalos existentes.

3.2 PRÉ-PROCESSAMENTO DE DADOS

A etapa inicial da metodologia envolve a importação e o pré-processamento dos dados. Esse processo é crucial para garantir a qualidade e a consistência dos dados usados nas análises subsequentes. Utilizando o código do Apêndice A, tem-se as primeiras etapas realizadas para o pré-processamento.

Este código lida com a importação inicial de dados e passos preliminares de limpeza, incluindo a renomeação de colunas e a conversão de informações de data e hora para análises futuras. Essa etapa é fundamental para assegurar que os dados estejam corretos e prontos para análises mais complexas, evitando assim erros ou imprecisões na interpretação dos resultados.

3.3 GRANULAÇÃO E TRANSFORMAÇÃO

Após a importação inicial dos dados, o conjunto de dados passa por granulação e transformação. Esta etapa envolve o refinamento dos dados por categorização e mapeamento de valores, utilizando o Apêndice B.

O código utiliza a transformação de colunas específicas e o tratamento de valores ausentes para preparar os dados para análise detalhada. A granulação e

transformação são passos críticos para estruturar os dados de uma maneira que facilite análises mais detalhadas e *insights* específicos.

3.4 CÁLCULO DO *AUTO DIE CHANGE* (ADC)

O ADC desempenha um papel impactante na eficiência do processo de manufatura e impacta significativamente o tempo de produção. O tempo de troca de matriz da peça que foi estampada para a próxima peça é importante e não é dado pelo *dataframe* inicial, sendo necessário o seu cálculo. Técnicas de filtragem de dados, para o cálculo e a análise do ADC, são aplicadas utilizando o código descrito no Apêndice C. O código filtra os dados com base em critérios específicos e calcula o tempo de ADC e o tempo de produção efetivo. Nessa parte que a utilização do Python é de extrema importância, pois a organização em um *dataframe* possibilita cálculos complexos com base no *timestamp* de linha, permitindo uma filtragem avançada, isolando os aspectos mais relevantes dos dados, essencial para a identificação de tendências ou padrões significativos.

3.5 CÁLCULO DO OEE

O cálculo da Efetividade Global dos Equipamentos (OEE) é o foco principal do estudo. O código Python contido no Apêndice D demonstra como os componentes do OEE são computados e a abordagem usada. É vital fazer algumas concessões, pois especificamente para este trabalho, foi considerada a Qualidade como sendo 1 para todos os cálculos, já que não foi possível o acesso as informações de qualidade vindas do PLC.

3.6 APRENDIZADO DE MÁQUINA

Para o aprendizado de máquina, seguiremos o seguinte conceito abaixo:

- Leitura dos Dados: Carregar os dados.
- Pré-processamento dos Dados: Limpar e preparar os dados para modelagem.
- Seleção de Características: Escolher as variáveis (em ingles *features*) relevantes para a previsão do OEE.
- Divisão dos Dados: Separar os dados em conjunto de treinamento e teste.
- Escolha do Modelo: Selecionar um modelo de *machine learning* adequado para regressão.
- Treinamento do Modelo: Treinar o modelo com o conjunto de dados de treinamento.
- Avaliação do Modelo: Avaliar a performance do modelo com o conjunto de dados de teste.
- Previsão: Usar o modelo treinado para prever o OEE do próximo lote.

O modelo escolhido esta inserido no pacote *SK-Learn* do Python e seja é uma analise simples, com uma previsão direta utilizando o modelo "*Fit()/Predict*".

3.7 ESTRUTURA DO *DATAFRAME* E EXEMPLO DE DADOS

A estrutura do *DataFrame* é essencial para compreender os dados utilizados nas análises do OEE. Abaixo está a estrutura detalhada do *dataframe* final desejado, com os tipos de dados e um exemplo de linha:

Tabela 1 – Estrutura e descrição do *DataFrame*

Nome da Coluna	Formato	Descrição e Utilização
Timestamp	datetime64[ns]	Data e hora da medição, usada para associar cada registro a um ponto específico no tempo.
Production Counter	int64	Contador de produção, representa o número total de peças produzidas até o momento da medição. Toda vez que o tipo de peça muda, o contador é zerado
ADC (min)	float64	Tempo, em minutos, que durou a troca de matriz, indicando tempo de inatividade/ajuste.
Current Part Type	int64	Tipo da peça atual sendo produzida, importante para análise de produção por tipo de peça.
Next Part Type	int64	Tipo da próxima peça a ser produzida, útil para planejamento e análise de sequenciamento de produção.
Total time of production (min)	float64	Tempo total de produção, em minutos, essencial para calcular a eficiência e a produtividade.
Effective Prod Time	float64	Tempo efetivo de produção, considerando o tempo de produção menos o tempo de ADC.
SPM	float64	Batidas por minuto, uma medida de produtividade e eficiência da linha de produção.
Performance	float64	Indicador de desempenho, calculado como parte da análise OEE.
Availability	float64	Disponibilidade do equipamento, outro componente crucial na análise OEE.
OEE	float64	Eficácia Global dos Equipamentos, métrica chave para avaliar a eficiência da produção.
Shift	object	Turno de trabalho em que os dados foram registrados, relevante para análises de desempenho por turno.

Fonte: Autor

Um exemplo de linha deste *DataFrame* desejado é apresentado a seguir:

```
"Timestamp": "2022-04-01 03:42:18.260",  
"Production Counter": 741,  
"ADC (min)": 7.103217,  
"Current Part Type": 28,  
"Next Part Type": 36,  
"Total time of production (min)": 104.4745,  
"Effective Prod Time": 97.371283,  
"SPM": 7.610047,  
"Performance": 1.064342,  
"Availability": 0.93201,  
"OEE": 0.991978,  
"Shift": "Third shift"
```

A estrutura detalhada do *DataFrame* e o exemplo de linha oferecem uma visão clara dos dados coletados para a análise. A *df* possui como pontos-chave o contador de produção, cálculos de tempo de produção, desempenho, disponibilidade e valores calculados do OEE, juntamente com o turno em que essas métricas foram registradas. Compreender esta estrutura de dados é vital para interpretar os resultados da análise e a eficácia no processo de produção.

4 RESULTADOS

4.1 IMPORTAÇÃO E TRATAMENTO DOS DADOS

Os dados utilizados foram recebidos diretamente do CLP, entre os períodos de Abril de 2022 até Setembro de 2023, os dados estão organizados em três colunas:

- **DB:** Trata-se de uma área de memória que pode ser organizada para armazenar diversas informações do seu processo. No nosso caso, é a porta que recebeu o *Value*. Cada *data block* tem um significado, o qual está explicitado no código presente no Apêndice A
- **Dt_Time:** O *datetime* em que o *Value* foi armazenado no *data block*
- **Value:** Valor, podendo ser caracteres, números inteiros ou *floats*

Abaixo esta uma síntese do resultado do pré-processamento, depois de executado o código do Apêndice A:

```
Index(['DB', 'Dt_Time', 'Value'], dtype='object')
      DB      Value
Dt_Time
2022-04-01 00:00:01.653    Accumulated day  5938.00
2022-04-01 00:00:02.590  Production Counter   475.00
2022-04-01 00:00:08.637    Accumulated day  5939.00
2022-04-01 00:00:09.623          SPM Line     8.22
2022-04-01 00:00:10.687  Production Counter   476.00
...
2023-09-20 06:05:48.820  Production Counter   635.00
2023-09-20 06:09:31.983    Accumulated day    16.00
2023-09-20 06:09:34.983  Production Counter   636.00
2023-09-20 06:09:39.013    Accumulated day    17.00
2023-09-20 06:09:41.997  Production Counter   637.00
```

```
[4962032 rows x 2 columns]
```

Nota-se o pivotamento bem sucedido porém, o *dataset* não pode ser utilizado de maneira tão simples, dado o tamanho e memória computacional de 4,962,032 linhas.

4.2 PIVOTAMENTO E PREPARAÇÃO

Concentrando a análise na importância do pré-processamento e organização dos dados para a efetiva análise de eficiência no processo de produção automotiva. Foi utilizado o código do Apêndice B, onde continuou-se o processo de preparação de dados com a inclusão de análises e visualizações preliminares. Este *script* define parâmetros ideais de produção e tempo de trabalho, estabelecendo um padrão de referência.

4.3 CÁLCULO DO ADC

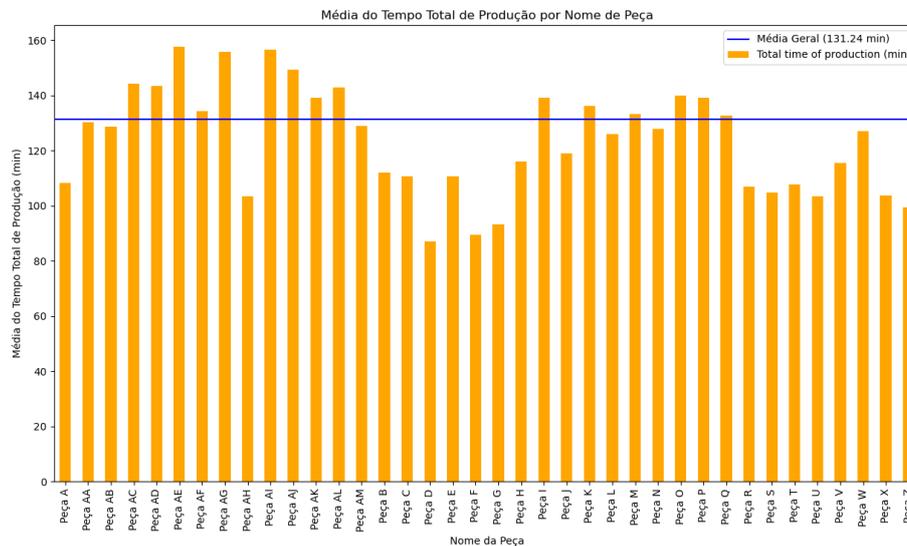
Utilizando o código do Apêndice C, realizamos o cálculo do tempo de ADC

A Figura 10 oferece uma visão detalhada do tempo médio de produção para diferentes tipos de peças. Esta análise é crucial para identificar possíveis gargalos no processo de produção, o tempo médio de referência em comparação com o atual geral. Peças com tempo de produção elevado podem indicar a necessidade de otimização de processos ou realocação de recursos para melhorar a eficiência geral.

Como apresentado no gráfico 10, o tempo médio de produção obtido para esta estamperia é de 131 minutos, Com a peça D apresentando o menor valor de produção, de aproximadamente 85 minutos, a peça AE obteve o maior valor médio, sendo aproximadamente 160 minutos.

Na Figura 11, é apresentada a relação entre o *Auto Die Change* (ADC) e o tempo total de produção. Interpretando a figura, não nota-se relação entre o

Figura 10 – Média do Tempo Total de Produção por Tipo de Peça.

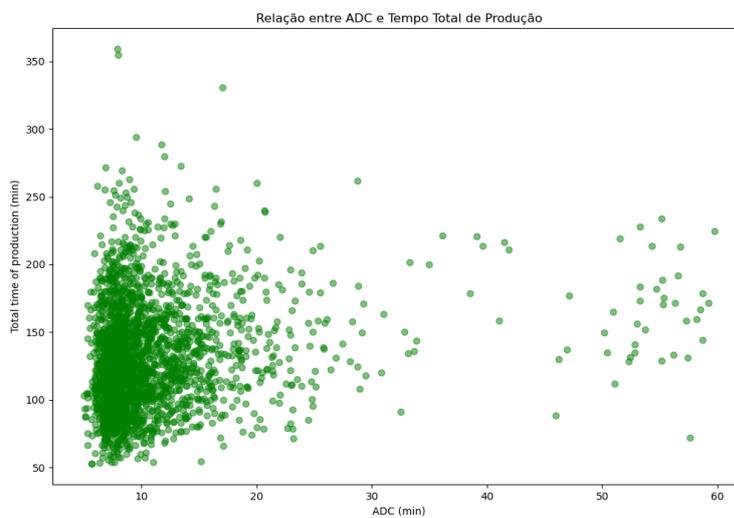


Fonte: Autor

tempo de ADC e a quantidade de peças produzidas pois o gráfico é disperso, não apresentando uma relação direta entre as duas.

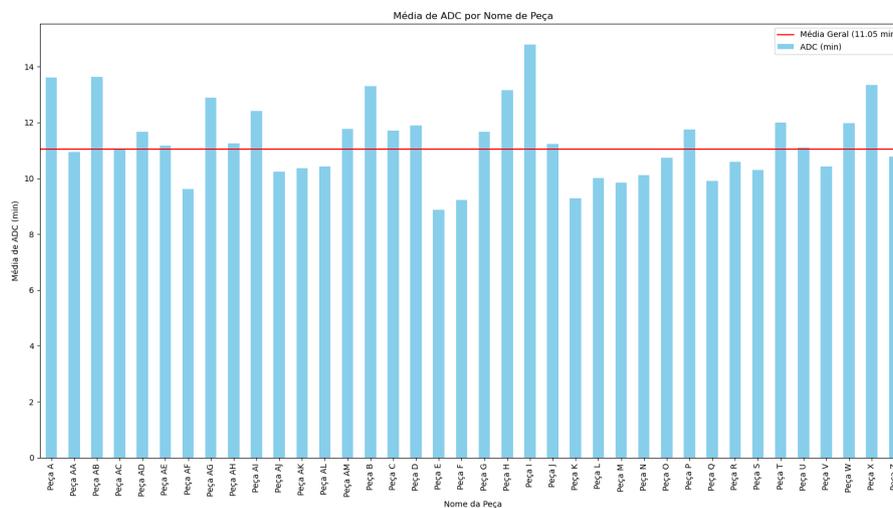
Por fim, a Figura 12 ilustra a média no tempo de ADC para cada tipo de peça produzida. Este dado é essencial para identificar quais peças têm a troca de matriz abaixo do tempo estipulado e quão desfazado está esse valor, o que pode ser um indicador de complexidade ou desafio na produção.

Figura 11 – Relação entre ADC e Tempo Total de Produção.



Fonte: Autor

Figura 12 – Média de ADC por Tipo de Peça.



Fonte: Autor

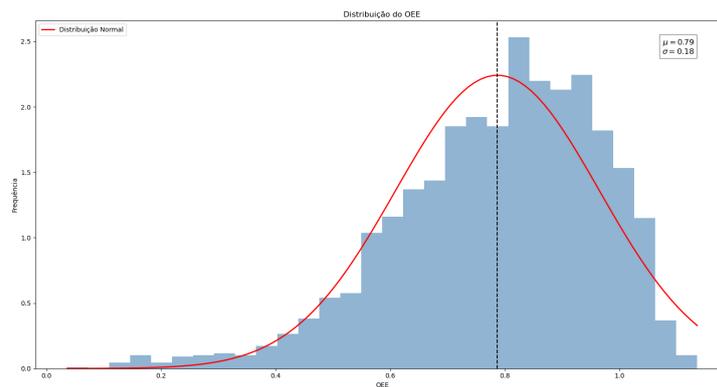
A média calculada foi de 11 minutos no geral, com a peça F apresentando o

menor tempo médio de troca (aproximadamente 9 minutos) e a peça I a maior média, com aproximadamente 15 minutos. A análise dessa figura permite a identificação de áreas que podem se beneficiar de melhorias no processo de ADC, visando a otimização do tempo e recursos na linha de produção.

4.4 CÁLCULO DO OEE

Utilizando o código contido no Apêndice D, foi calculada uma distribuição normal do OEE incluindo todos os lotes de peças produzidos, como mostra a Figura 13.

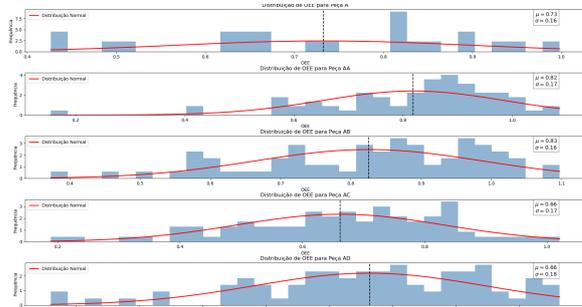
Figura 13 – Média geral do OEE.



Fonte: Autor

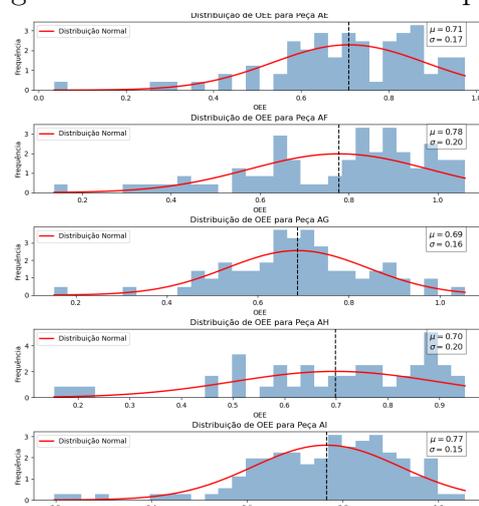
Após o cálculo geral, focou-se no cálculo de cada tipo de peça individual, mas agrupados para uma melhor visualização geral, dado o grande número de tipos de peças, gerando as Figuras 14, 15, 16, 17, 18, 19, 20 e 21:

Figura 14 – OEE individuais - Grupo 1



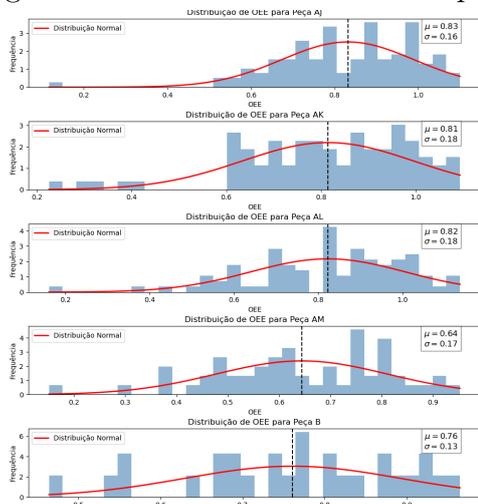
Fonte: Autor

Figura 15 – OEE individuais - Grupo 2



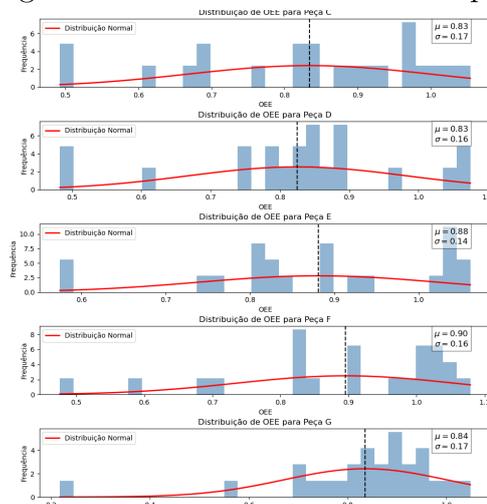
Fonte: Autor

Figura 16 – OEE individuais - Grupo 3



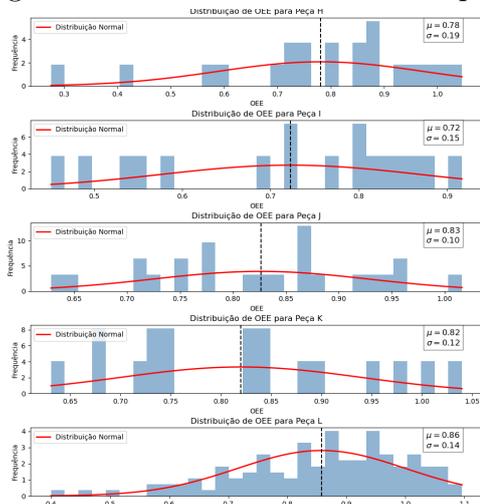
Fonte: Autor

Figura 17 – OEE individuais - Grupo 4



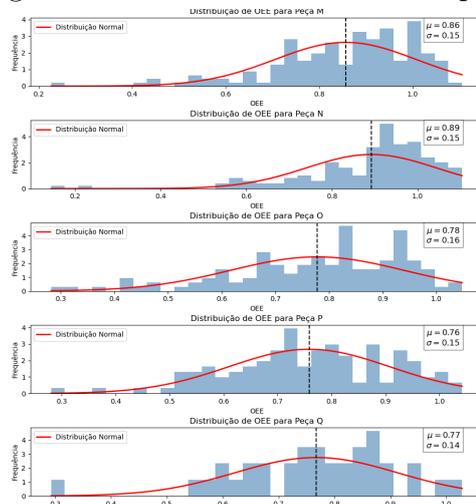
Fonte: Autor

Figura 18 – OEE individuais - Grupo 5



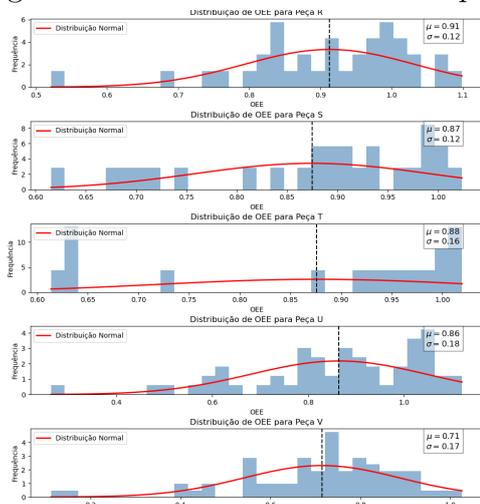
Fonte: Autor

Figura 19 – OEE individuais - Grupo 6



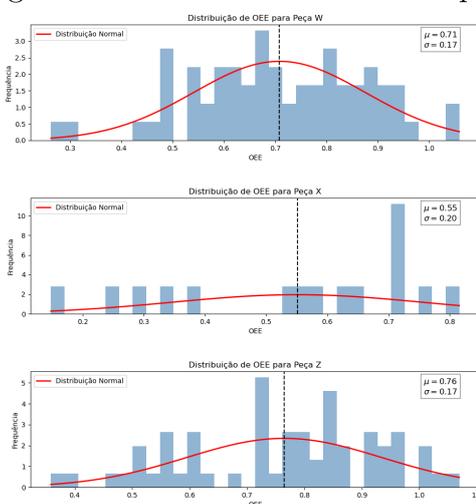
Fonte: Autor

Figura 20 – OEE individuais - Grupo 7



Fonte: Autor

Figura 21 – OEE individuais - Grupo 8



Fonte: Autor

Analisando individualmente, tem-se que os valores de OEE variam para

cada tipo de peça, com médias (μ) indo de valores mais baixos como 0,55 para a peça X até valores mais altos como 0,91 para a peça R. Os desvios padrão (σ) também variam, indicando que algumas distribuições são mais estreitas e outras mais dispersas.

Ao comparar os gráficos, percebe-se que há variações significativas no OEE entre os diferentes tipos de peças. Algumas peças, como R, S, T, U e F, têm valores médios de OEE mais altos e distribuições mais concentradas em torno da média, indicando uma produção consistentemente eficiente. Por outro lado, peças como X e W apresentam médias mais baixas de OEE e distribuições mais amplas, sugerindo maior variabilidade na eficiência da produção.

Portanto, os gráficos fornecem uma visualização útil para entender como o OEE varia entre diferentes tipos de peças em um processo de produção. Eles são instrumentos valiosos para identificar onde as melhorias são necessárias e para realizar *benchmarking* entre as peças. As diferenças observadas nos gráficos podem ser devido a vários fatores, como a complexidade da fabricação das peças, a eficiência das máquinas utilizadas, ou a habilidade dos operadores, e podem indicar oportunidades de otimização de processos.

4.5 APRENDIZADO DE MÁQUINAS

Para o aprendizado de máquina, contido no Apêndice E a previsão do *OEE* para 10 lotes futuros, exemplificados pela Tabela 2:

Tabela 2 – Dados utilizados para a previsão do OEE

Production Counter	ADC (min)	Current Part Type	Next Part Type	Total time of production (min)	Effective Prod Time	SPM	Performance	Availability
700	10.089167	48	28	155.013217	144.924050	4.830116	0.675541	0.934914
935	12.238383	50	42	141.450600	129.212217	7.236158	1.012050	0.913479
616	33.834950	54	28	143.484450	109.649500	5.617901	0.785720	0.764191
560	12.419233	39	25	95.993950	83.574717	6.700591	0.937146	0.870625

Continua na próxima página

Tabela 2 – (Continuação) Dados utilizados para a previsão do OEE

Production Counter	ADC (min)	Current Part Type	Next Part Type	Total time of production (min)	Effective Prod Time	SPM	Performance	Availability
690	16.116717	29	52	106.948717	90.832000	7.596442	1.062439	0.849304
731	11.864283	47	46	131.518117	119.653833	6.109290	0.854446	0.909790
627	8.237167	48	48	103.861600	95.624433	6.556902	0.917049	0.920691
418	12.918167	32	45	93.020717	80.102550	5.218311	0.729834	0.861126
790	8.089733	38	28	123.473500	115.383767	6.846717	0.957583	0.934482
417	23.203267	51	37	71.558500	48.355233	8.623679	1.206109	0.675744

Fonte: Autor

O modelo da Tabela 2 foi realizado usando o método de *machine learning* que compara um dos dados de teste (modelo 80/20) com o previsto, comparando o real com a predição do modelo. O *Mean Squared Error (MSE)* do modelo foi aproximadamente $9.93 * 10^{-5}$, quanto menor o valor, mais acurativo é o modelo treinado, indicando o desempenho do modelo na previsão do OEE com os dados de teste.

O *MSE* mede a média dos quadrados dos erros, o *MSE* é sempre positivo ou 0. Não existe um *target* para o *MSE*, apenas é um indicador do quão o modelo está adaptado aos dados utilizados. O modo de cálculo está representado na equação 5:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (5)$$

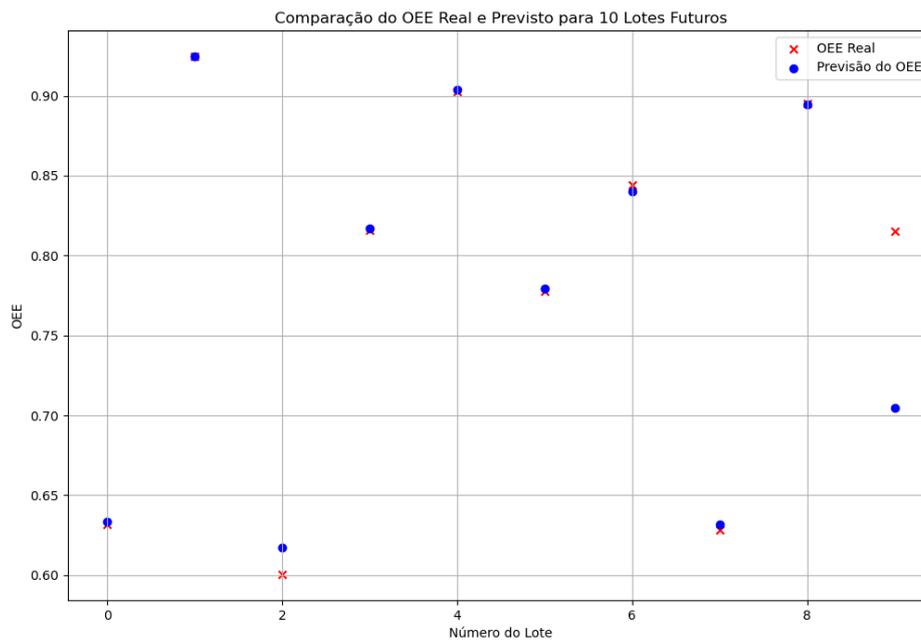
Onde:

- *MSE* representa o *Mean Squared Error*.
- n é o número de observações no conjunto de dados.
- y_i são os valores reais das observações.
- \hat{y}_i são os valores previstos (ou estimados) das observações.

Na figura 22, cada ponto representa a previsão do OEE para um lote futuro, utilizando um conjunto de dados fictício gerado a partir dos dados de teste. Essas previsões fornecem uma estimativa de como o OEE pode variar nos próximos lotes com base nas características fornecidas ao modelo. A precisão dessas previsões está

diretamente relacionada à qualidade e relevância dos dados de entrada, bem como à adequação do modelo escolhido para os padrões presentes nos dados. O código para a predição está apresentado no Apêndice E.

Figura 22 – Comparação da predição e do resultado real dos OEE



Fonte: Autor

O modelo é considerado satisfatório, pois mesmo sendo um modelo simples, sua proposta inicial é apenas ilustrar o potencial do aprendizado de máquina quando em conjunto com a análise de dados.

5 CONCLUSÃO

Este trabalho apresenta uma análise aprofundada da aplicação do modelo de OEE (*Overall Equipment Effectiveness*) em uma estamperia automotiva, complementado por técnicas de aprendizado de máquina para previsão de eficiência em lotes futuros. Os resultados obtidos demonstram a relevância e eficácia do modelo OEE, não apenas como uma ferramenta de medição, mas também como um indicador estratégico para a otimização do processo produtivo, para o caso deste estudo, um OEE de 79% é considerado satisfatório, vide 2.3.4.

Temos também a análise aprofundada de cada tipo de lote de peças com o uso das distribuições normais de todos os diferentes tipos produzidos, para uma análise mais individual. Foi observado que com uma quantidade maior de dados, a dispersão e flutuação das frequências dos gráficos individuais seja mais constante, trazendo resultados mais próximos aos reais.

No entanto, este estudo não está isento de limitações. A utilização da qualidade como sendo 100% para o cálculo eleva muito o valor final do OEE. Portanto, recomenda-se cautela na generalização das conclusões para diferentes contextos industriais.

Para o aprendizado de máquina, o resultado do *MSE* é considerado bom pelo Autor, porém, é apenas um início dentro de uma vasta gama de métodos e possibilidades que essa ferramenta traz. Fica a atenção para o *overfit*, que foi evitado durante o código, mas mesmo assim é um cuidado a ser tomado para futuras análises.

Para pesquisas futuras, sugere-se a exploração de diferentes abordagens de aprendizado de máquina e a inclusão de variáveis adicionais que possam impactar a eficiência da produção. Além disso, a aplicação do modelo OEE em diferentes setores industriais pode fornecer leituras adicionais sobre sua adaptabilidade e eficácia em diversos ambientes de produção.

A integração do modelo OEE com técnicas de aprendizado de máquina apresenta uma nova direção para a pesquisa na área, abrindo caminho para futuros desenvolvimentos que podem transformar a maneira como as indústrias abordam a

eficiência e sustentabilidade da produção. A otimização da produção não é apenas uma questão de maximizar o lucro, mas também de garantir a sustentabilidade e a resiliência dos processos produtivos em um mercado cada vez mais competitivo e dinâmico.

REFERÊNCIAS

- BUSSO, C. M.; MIYAKE, D. I. Análise da aplicação de indicadores alternativos ao overall equipment effectiveness (oeo) na gestão do desempenho global de uma fábrica. **Production**, SciELO Brasil, v. 23, p. 205–225, 2013.
- CADAVID, J. P. U. et al. Machine learning applied in production planning and control: a state-of-the-art in the era of industry 4.0. **Journal of Intelligent Manufacturing**, Springer, v. 31, p. 1531–1558, 2020.
- GEN, Y.; YUNONG, W. Progressive stamping process and die design of high strength steel automobile structural parts. **Journal of Physics: Conference Series**, IOP Publishing, v. 1605, n. 1, p. 012063, aug 2020. Disponível em: <<https://dx.doi.org/10.1088/1742-6596/1605/1/012063>>.
- GROOTE, P. D. Maintenance performance analysis: a practical approach. **Journal of Quality in Maintenance Engineering**, MCB UP Ltd, v. 1, n. 2, p. 4–24, 1995.
- HANSEN, R. C. **Overall equipment effectiveness: a powerful production/maintenance tool for increased profits**. [S.l.]: Industrial Press Inc., 2001.
- HASSANI, I. E.; MAZGUALDI, C. E.; MASROUR, T. Artificial intelligence and machine learning to predict and improve efficiency in manufacturing industry. **ArXiv**, abs/1901.02256, 2019.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. The elements of statistical learning. springer series in statistics. **New York, NY, USA**, 2001.
- KUMAR, R. Future for scientific computing using python. **International Journal of Engineering Technologies and Management Research**, v. 2, n. 1, p. 30–41, Jul. 2015. Disponível em: <https://www.granthaalayahpublication.org/ijetmr-ojms/ijetmr/article/view/04_IJETMR15_A07_22>.
- LJUNGBERG, Ö. Measurement of overall equipment effectiveness as a basis for tpm activities. **International Journal of Operations & Production Management**, MCB UP Ltd, v. 18, n. 5, p. 495–507, 1998.
- MALEKI, F. et al. Overview of machine learning part 1: fundamentals and classic approaches. **Neuroimaging Clinics**, Elsevier, v. 30, n. 4, p. e17–e32, 2020.
- MCKINNEY, W. et al. pandas: a foundational python library for data analysis and statistics. **Python for high performance and scientific computing**, Seattle, v. 14, n. 9, p. 1–9, 2011.

- MJIMER, I.; AOULA, E.-S.; ACHOUYAB, E. H. Contribution of machine learning in continuous improvement processes. **Journal of Quality in Maintenance Engineering**, Emerald Publishing Limited, v. 29, n. 2, p. 553–567, 2022.
- NAKAJIMA, S. Tpm tenkai. **Japan Institute of Plant Maintenance, Tokyo**, 1982.
- NAKAJIMA, S. Introduction to tpm total productive maintenance, productivity-press. **Inc. Cambridge. Massachusetts**, 1998.
- NONGTHOMBAM, K.; SHARMA, D. Data analysis using python. **International Journal of Engineering Research & Technology (IJERT)**, v. 10, n. 7, 2021.
- NUNES, D.; CAMPILHO, R.; SILVA, F. Design of a transfer system for the automotive industry. **Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering**, v. 236, p. 2044 – 2055, 2022.
- PEDREGOSA, F. et al. Scikit-learn: Machine learning in python. **the Journal of machine Learning research**, JMLR. org, v. 12, p. 2825–2830, 2011.
- RABELO, G. M. **Estudo e Análise do Indicador Overall Equipment Effectiveness of a Manufacturing Line Aplicado em uma Empresa de Autopeças**. Tese (Doutorado) — [sn], 2016.
- RASCHKA, S.; PATTERSON, J.; NOLET, C. Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. **Information**, v. 11, n. 4, 2020. ISSN 2078-2489. Disponível em: <<https://www.mdpi.com/2078-2489/11/4/193>>.
- SAHOO, K. et al. Exploratory data analysis using python. **International Journal of Innovative Technology and Exploring Engineering**, v. 8, n. 12, p. 4727–4735, 2019.
- SANTOS, P. V. S. Aplicação do indicador overall equipment effectiveness (oee): um estudo de caso numa retífica e oficina mecânica. **Brazilian journal of production engineering**, v. 4, n. 3, p. 1–18, 2018.
- SHINKEVICH, A. et al. Reserves for improving the efficiency of petrochemical production on the basis of “industry 4.0”. **E3S Web of Conferences**, 2019.
- SRINATH, K. Python—the fastest growing programming language. **International Research Journal of Engineering and Technology**, v. 4, n. 12, p. 354–357, 2017.
- TAYAL, A. et al. Effectiveness improvement in manufacturing industry; trilogy study and open innovation dynamics. **Journal of Open Innovation: Technology, Market, and Complexity**, 2020.

TEREHOV, M.; ZAIKIN, V.; AVERCHENKOV, A. Increasing industrial efficiency based on developing an automated production planning system. v. 2021, p. 49–57, 2021.

TSAROUHAS, P. H. Overall equipment effectiveness (oe) evaluation for an automated ice cream production line: A case study. **International Journal of Productivity and Performance Management**, Emerald Publishing Limited, v. 69, n. 5, p. 1009–1032, 2020.

VIJAYAKUMAR, S.; GAJENDRAN, S. Improvement of overall equipment effectiveness (oe) in injection moulding process industry. **IOSR J Mech Civil Eng**, v. 2, n. 10, p. 47–60, 2014.

YAZDI, P. G.; AZIZI, A.; HASHEMIPOUR, M. An empirical investigation of the relationship between overall equipment efficiency (oe) and manufacturing sustainability in industry 4.0 with time study approach. **Sustainability**, v. 10, n. 9, 2018. ISSN 2071-1050. Disponível em: <<https://www.mdpi.com/2071-1050/10/9/3031>>.

YOGESH, M.; PRABAGARAN, S. Measures for enhancing productivity and efficiency of small and medium scale industry by lean implementation. **Indian journal of science and technology**, v. 10, p. 1–5, 2017.

YU, L. Feeding mechanisms in the auto-body stamping production line: A review of common research. **Journal of The Institution of Engineers (India): Series C**, Springer, v. 104, n. 2, p. 439–448, 2023.

ZENNARO, I. et al. Micro downtime: Data collection, analysis and impact on oe in bottling lines the san benedetto case study. **International Journal of Quality & Reliability Management**, Emerald Publishing Limited, v. 35, n. 4, p. 965–995, 2018.

ZUCCOTTO, P. A simulation modelling approach to improve the oe of a bottling line. 2019.

ÇINAR, Z. M. et al. Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. **Sustainability**, v. 12, n. 19, 2020. ISSN 2071-1050. Disponível em: <<https://www.mdpi.com/2071-1050/12/19/8211>>.

APÊNDICE A – IMPORTAÇÃO E TRATAMENTO INICIAL DOS DADOS

O código a seguir foi utilizado para ler e padronizar os dados dentro de alguns parâmetros, como por exemplo, excluir os domingos do *timestamp* e organiza-los em ordem cronológica.

```
# Importações necessárias
import pandas as pd

df = pd.read_csv('Dataframes\df.csv')

# Converte a coluna 'Dt_Time' para o formato de data e hora
df['Dt_Time'] = pd.to_datetime(df['Dt_Time'], format='mixed')

# Filtra as linhas onde o dia da semana não é domingo (6 representa
↪ domingo)
df = df[df['Dt_Time'].dt.weekday != 6]

# Filtra as linhas onde o dia da semana não é sábado (5 representa
↪ sábado) e o horário é posterior a 06:05:00
df = df[~((df['Dt_Time'].dt.dayofweek.isin([5,6])) &
↪ (df['Dt_Time'].dt.time >= pd.Timestamp('06:05:00').time()))]

# Define a coluna 'Dt_Time' como índice do DataFrame e ordena as linhas
↪ em ordem crescente de data e hora
df.set_index('Dt_Time', inplace = True)
df.sort_index(ascending=True, inplace = True)
```

APÊNDICE B – Limpeza e Pivotamento

Código utilizando para limpar e destacar as informações relevantes para a análise

```
# Importações necessárias
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Carregando o dataframe a partir de um arquivo pickle
df = pd.read_pickle('df.pkl')

# Filtrando e manipulando o dataframe
Db_label = ['Production Counter', 'Part Type']
df_PP = df.loc[df['DB'].isin(Db_label)]

# Pivotando o dataframe para ter 'Production Counter' e 'Part Type' como
↪ colunas
df_PP = df_PP.pivot(columns='DB', values='Value')

# Preenchendo valores ausentes com o último valor válido (forward fill)
df_PP.replace(to_replace=np.nan, method='ffill', inplace=True)

# Removendo linhas com valores nulos
df_PP = df_PP.dropna()

# Convertendo as colunas para tipo numérico
df_PP = df_PP.apply(pd.to_numeric)

# Filtrando e manipulando outra parte do dataframe para criar um
↪ dicionário de correspondência entre 'Part Type' e 'Part Name'
temp_label = ['Part Name', 'Part Type']
temp = df.loc[df['DB'].isin(temp_label)]
```

```

temp = temp.pivot(columns='DB', values='Value')
temp['Part Type'] = pd.to_numeric(temp['Part Type'])
part_dict = dict(zip(temp['Part Type'], temp['Part Name'].shift()))

# Removendo chaves com valores nulos
part_dict = {k: part_dict[k] for k in part_dict if not pd.isna(k)}

# Identificando mudanças de peça com base em resets no contador
counter_resets = df_PP['Production Counter'] == 0
df_PP['Part Change'] = counter_resets.shift(-1)

# Criando um novo dataframe para armazenar os dados após a identificação
↳ de mudanças de peça
df_ADC = df_PP.loc[~(df_PP['Production Counter'] == 0)]
df_ADC['Part Change'][-1] = False
df_ADC = df_ADC[1:]

# Análise da produção diária em comparação com os parâmetros
Db_label = ['Accumulated day']

# Filtrando dados relacionados à contagem acumulada diária
df_Prod_count = df.loc[df['DB'].isin(Db_label)]
df_Prod_count = pd.to_numeric(df_Prod_count["Value"])

# Redimensionando a série temporal para uma resolução diária
df_Prod_count = df_Prod_count.resample('D').max()
df_Prod_count.index = pd.to_datetime(df_Prod_count.index)
df_Prod_count = df_Prod_count.to_frame('Valor')

# Exibindo o dataframe resultante
print(df_Prod_count.head())

# Validando a contagem de produção em relação ao valor ideal
df_Prod_count["validation"] = df_Prod_count > IProd_per_day
df_Prod_count.columns = ['Valor', 'Validacao']

```

```
df_Prod_count.Validacao.value_counts()
df_Prod_count["Cor"] = df_Prod_count["Valor"] > IProd_per_day
df_Prod_count["Cor"] = df_Prod_count["Cor"].astype(int)
mydict = ["Red", "Blue"]
mydict = dict(enumerate(mydict))
df_Prod_count['Cor'] = df_Prod_count['Cor'].map(mydict)
df_Prod_count.index[df_Prod_count['Validacao'] == True].tolist()
```

APÊNDICE C – Calculo do tempo de troca de matriz

Calculo do tempo de troca de matriz para cada lote produzido

```

import pandas as pd # Importa a biblioteca pandas como "pd"
import matplotlib.pyplot as plt # Importa a biblioteca
↳ matplotlib.pyplot como "plt"
import numpy as np # Importa a biblioteca numpy como "np"

df_ADC = pd.read_pickle('df_ADC.pkl') # Lê o arquivo 'df_ADC.pkl' como um
↳ dataframe e o armazena em 'df_ADC'

# Define uma função para calcular a diferença de tempo
def calculate_time_diff(df):
    part_change_mask = df['Part Change'] == True # Cria uma máscara para
↳ as linhas em que 'Part Change' é True
    part_change_rows = df[part_change_mask].copy() # Cria uma cópia das
↳ linhas onde 'Part Change' é True
    part_change_rows['Time Diff'] =
↳ (part_change_rows.index.to_series().shift(-1) -
↳ part_change_rows.index).dt.total_seconds() / 60.0 # Calcula a
↳ diferença de tempo e a adiciona como uma coluna 'Time Diff'
    return part_change_rows.dropna().reset_index(drop=True).loc[:,
↳ ['Production Counter', 'Part Type', 'Time Diff']] # Retorna as
↳ colunas específicas do dataframe resultante

result2 = calculate_time_diff(df_ADC) # Chama a função e armazena o
↳ resultado em 'result2'

# ADC
# Cria dois arrays numpy para armazenar os valores de diferença de tempo
↳ e contador de produção

```

```

time_difference = np.array([], dtype=np.float64)
production_counter = np.array([], dtype=np.int64)
part_type = [] # Cria uma lista vazia para armazenar os tipos de peças
timestamp_store = [] # Cria uma lista vazia para armazenar os carimbos
↳ de data/hora

# Loop através do dataframe
for i in range(len(df_ADC) - 1):
    # Se a linha atual tiver 'Part Change' como True
    if df_ADC.iloc[i]['Part Change'] == True:
        # Calcula a diferença de tempo entre a linha atual e a próxima
        ↳ linha
        time_difference = np.append(time_difference, (df_ADC.index[i +
        ↳ 1] - df_ADC.index[i]).total_seconds() / 60)
        # Armazena o valor do contador de produção da linha atual
        production_counter = np.append(production_counter,
        ↳ df_ADC.iloc[i]['Production Counter'])
        part_type.append(df_ADC.iloc[i]['Part Type']) # Adiciona o tipo
        ↳ de peça à lista 'part_type'
        timestamp_store.append(df_ADC.index[i]) # Adiciona o carimbo de
        ↳ data/hora à lista 'timestamp_store'

# Cria um novo dataframe com os resultados
df_ADC = pd.DataFrame({
    'Timestamp': timestamp_store, # Adiciona o índice de carimbo de
    ↳ data/hora original
    'Production Counter': production_counter,
    'ADC (min)': time_difference,
    'Current Part Type': part_type,
})

df_ADC['Next Part Type'] = df_ADC['Current Part Type'].shift(-1) # Cria
↳ uma coluna 'Next Part Type' com os valores deslocados

```

```

df_ADC["Total time of production (min)"] = result2['Time Diff'].shift()
↳ # Cria uma coluna 'Total time of production (min)' com os valores de
↳ 'result2' deslocados
df_ADC["ADC (min)"] = df_ADC["ADC (min)"].shift() # Desloca os valores
↳ da coluna 'ADC (min)'
df_ADC.dropna(inplace=True) # Remove linhas com valores ausentes

# Converte a coluna de tempo para o formato datetime
time_diff = df_ADC['ADC (min)']

# Relaciona a variável com a coluna
Total_Time = df_ADC['Total time of production (min)']

# Relaciona a variável com a coluna
ProductionCounter = df_ADC['Production Counter']

# Cálculos para os gráficos

# Média de ADC por nome de peça
media_ADC_por_nome = df_ADC.groupby('Part Name')['ADC (min)'].mean()

# Média geral de ADC
media_ADC_geral_atualizada = df_ADC['ADC (min)'].mean()

# Média do tempo total de produção por nome de peça
media_tempo_total_por_nome = df_ADC.groupby('Part Name')['Total time of
↳ production (min)'].mean()

# Média geral do tempo total de produção
media_tempo_total_geral_atualizada = df_ADC['Total time of production
↳ (min)'].mean()

```

APÊNDICE D – Calculo do OEE

Calculo efetivo do OEE para cada lote estampado na montadora

```

# Importações
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import math
from scipy.stats import norm

df_ADC = pd.read_pickle('df_ADC_parsed.pkl')

# Cálculo do OEE

# Filtrando o DF

# Relacionar variável à coluna
TotalTimeProduction = df_ADC['Total time of production (min)']
df_ADC['Effective Prod Time'] = df_ADC['Total time of production
↪ (min)'] - df_ADC['ADC (min)']
df_ADC['SPM'] = df_ADC['Production Counter'] / df_ADC['Effective Prod
↪ Time']

daily_changes = df_ADC.groupby(pd.Grouper(key='Timestamp', freq='D'))['Next
↪ Part Type'].count()

# OEE Geral

# Calcular Disponibilidade
total_production_time = df_ADC['Total time of production (min)'].sum()
total_planned_production_time = df_ADC['Effective Prod Time'].sum()
availability = total_planned_production_time / total_production_time

```

```

# Calcular Desempenho
ideal_cycle_time = 60/SPM_Ideal
total_parts_produced = df_ADC['Production Counter'].sum()
total_planned_production_time_seconds = total_planned_production_time *
↪ 60
ideal_production_time = total_parts_produced * ideal_cycle_time
performance = ideal_production_time /
↪ total_planned_production_time_seconds

# Calcular Qualidade
quality = Quality_Parameter

# Calcular OEE
oee = availability * performance * quality

print("OEE geral: {:.2%}".format(oee))

# OEE por Peça Produzida
# Calcular o OEE para cada linha
perf = []
avail = []
oee = []

for i, row in df_ADC.iterrows():
    # Calcular Disponibilidade
    total_production_time = row['Total time of production (min)']
    total_planned_production_time = row['Effective Prod Time']
    availability = total_planned_production_time /
    ↪ total_production_time if total_production_time != 0 else 0

    # Calcular Desempenho
    ideal_cycle_time = 60 / SPM_Ideal
    actual_cycle_time = 60 / row['SPM'] if row['SPM'] != 0 else 0
    performance = ideal_cycle_time / actual_cycle_time if
    ↪ actual_cycle_time != 0 else 0

```

```
# Calcular Qualidade
quality = Quality_Parameter

# Calcular OEE
row_oeo = availability * performance * quality if (availability *
↪ performance * quality) != 0 else 0
avail.append(availability)
perf.append(performance)
oeo.append(row_oeo)

# Adicionar a coluna OEE ao DataFrame
df_ADC['Performance'] = perf
df_ADC['Availability'] = avail
df_ADC['OEE'] = oeo

# Calcular a média e desvio padrão da coluna OEE
mean_OEE = df_ADC['OEE'].mean()
std_OEE = df_ADC['OEE'].std()
```

APÊNDICE E – Aprendizado de máquina simples

Aplicação do pacote *SK-learn* para a predição, treinamento e *fitting* do *dataframe*

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
import numpy as np
import matplotlib.pyplot as plt

# Leitura dos dados
df = pd.read_pickle('df_OEE.pkl')

# Conversão do Timestamp em características numéricas mais detalhadas
df['Hour'] = df['Timestamp'].dt.hour
df['Day'] = df['Timestamp'].dt.day
df['Weekday'] = df['Timestamp'].dt.weekday # Novo: Dia da semana
df['Month'] = df['Timestamp'].dt.month
df['Year'] = df['Timestamp'].dt.year

# Removendo colunas não necessárias
df = df.drop(['Timestamp', 'Shift'], axis=1) # Mantendo 'Current Part
↳ Type' e 'Next Part Type'

# Preenchendo valores ausentes
df.fillna(method='ffill', inplace=True)

# Seleção de Características
features = df.drop('OEE', axis=1)
target = df['OEE']
```

```
# Divisão dos Dados
X_train, X_test, y_train, y_test = train_test_split(features, target,
↪ test_size=0.2, random_state=42)

# Escolha do Modelo - RandomForestRegressor com mais estimadores
model = RandomForestRegressor(n_estimators=200, random_state=42)

# Treinamento do Modelo
model.fit(X_train, y_train)

# Avaliação do Modelo
predictions = model.predict(X_test)
mse = mean_squared_error(y_test, predictions)
print(f'Mean Squared Error: {mse}')

# Previsão para 10 lotes aleatórios
next_batch_data = X_test.sample(10, random_state=42)
oe_predictions = model.predict(next_batch_data)
```