

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

Centro de Ciências Exatas e de Tecnologia

Departamento de Computação

Pedro Freire Baleeiro

Análise das Métricas SMART em HDDs para Modelos de Predição  
de Falha nos Discos usados em Data Centers para a Nuvem

**Trabalho de Conclusão de Curso**

Orientador: Hélio Crestana Guardia

São Carlos - SP

2024

## Resumo

O armazenamento de dados na nuvem é uma das principais atividades dos grandes provedores de infraestrutura em nuvem. Os objetos (dados não estruturados) compõem grande parte do que é guardado e os Discos Rígidos (HDs) ainda são a mídia mais utilizada para atingir quantidades significativas de armazenamento pelo melhor custo-benefício. Toda a infraestrutura do serviço na nuvem deve ser redundante, para que os dados e o mecanismo como um todo seja confiável, durável e disponível ao máximo. Dessa forma, um aspecto significativo e ainda não suficientemente explorado e que é tema deste trabalho é o de manutenção preditiva dos discos rígidos. Neste trabalho, os mecanismos por trás desse assunto são explicados por meio da comparação das estratégias de redundância (Replicação e *Erasure Coding*), do monitoramento de métricas SMART e da confiabilidade do sistema, do cálculo de Tempo de Vida Útil (RUL) dos dispositivos e do desenvolvimento de modelos de Aprendizado de Máquina para prever falhas iminentes em discos de um modelo. Ademais, é feita a replicação de um experimento descrito no artigo “Interpretable predictive maintenance for hard drives” [1] para um modelo de disco e intervalo diferentes. Por fim, considerações são reunidas sobre a efetividade desse modelo para tratar os dados em questão e prover maior durabilidade e confiabilidade para um sistema de armazenamento baseado em discos rígidos. Como resultado, busca-se fornecer um largo conhecimento dos assuntos que envolvem o tema deste trabalho.

Palavras-chave: Armazenamento; Nuvem; HDDs; Discos; Confiabilidade; Durabilidade; Redundância; SMART; RUL; Manutenção; Preditiva; AM; Modelo; Interpretável.

## Sumário

1.	<b>Introdução</b> .....	5
1.1.	Um Panorama da Computação em Nuvem e Data Centers	5
1.2.	Armazenamento de Dados	6
2.	<b>Discos e o Cenário de Armazenamento em Data Centers</b> .....	9
2.1.	Funcionamento dos HDDs e uso em Data Centers.	9
2.2.	Métricas de Confiabilidade (Reliability)	12
2.3.	Storage Pods e Vault Pods	18
3.	<b>Métricas SMART para HDs e Dataset da Backblaze</b> .....	20
3.1.	Visão geral do Dataset da Backblaze	21
3.2.	Atributos SMART com maior correlação com falhas em discos	23
3.3.	Dados Empíricos obtidos da Backblaze	26
4.	<b>Predição com Modelos de AM e Trabalhos Relacionados</b> .....	30
4.1.	Manutenção Preditiva	31
4.2.	Modelos de AM para Predição de Falhas em HDs	37
5.	<b>Desenvolvimento do Trabalho</b> .....	42
6.	<b>Resultados e Discussões</b> .....	53
7.	<b>Conclusões</b> .....	57
8.	<b>Referências</b> .....	61

## Capítulo 1. Introdução

### 1.1. Um Panorama da Computação em Nuvem e Data Centers

Computação em Nuvem é um tema bastante atual e relevante. No entanto, como foi dito por Bob Olwig do World Wide Technology, “Pergunte para 10 pessoas o que é Nuvem e você receberá 11 respostas” [2]. Isso indica o quanto essa área é nova, envolve assuntos extremamente complexos, com diversas frentes de atuação e trabalho para fazer com que a Nuvem seja o que ela é.

A Internet é construída quase que completamente em cima de nuvens e nuvens são construídas em cima de *data centers*. *Data centers* são construções que contêm uma grande quantidade de computadores, também conhecidos como servidores, que realizam os diversos serviços computacionais de uma ou mais empresas e usuários. A informação é processada e tornada disponível, o que inclui atividades essenciais que são o armazenamento, o gerenciamento e a disseminação dos dados entre esses computadores e outros.

Organizações precisam ter acesso a computadores servidores e muitas vezes elas constroem sua própria infraestrutura ou conjunto de servidores para compor um *data center*. No entanto, mesmo com apenas alguns racks de servidores já é necessário ter que manter os sistemas funcionais por quase todo o tempo, cuidando de questões operacionais, como energia e refrigeração, e lidando com aumento na demanda da empresa que utiliza a infraestrutura.

Essas tarefas são custosas, demandam pessoal qualificado e são constantes em servidores e *data centers*. Por esse motivo, é comum empresas procurarem uma forma diferente de realizarem seus serviços computacionais. Nessa parte, entra a utilização dos serviços de um provedor de nuvem pública. Nuvens como AWS, GCP, Azure e Magalu Cloud<sup>1</sup> são consideradas nuvens públicas pois apresentam uma arquitetura *multi-tenant* que, em tradução livre, significa atender a múltiplos locatários. Esse nome vem do fato de que os recursos computacionais são divididos entre os usuários de modo a apresentar maior eficiência enquanto tornam a nuvem escalável e apropriada para o uso de cada um.

Normalmente, nuvens públicas apresentam um modelo de “*pay as you go*” ou “*pay as you use*”, enquanto numa infraestrutura própria existem mais riscos pois é preciso realizar estimativas, planejamentos e grandes investimentos nas etapas iniciais para a construção,

---

<sup>1</sup> <https://aws.amazon.com/pt/> <https://cloud.google.com/>  
<https://azure.microsoft.com/pt-br/> <https://magalu.cloud/>

gastos contínuos e elevados com profissionais qualificados, novos equipamentos e manutenção. Apesar disso, a escalabilidade potencialmente infinita pode virar-se contra uma organização que contrata serviços de nuvens públicas, pois as contas do serviço também escalam de forma proporcional.

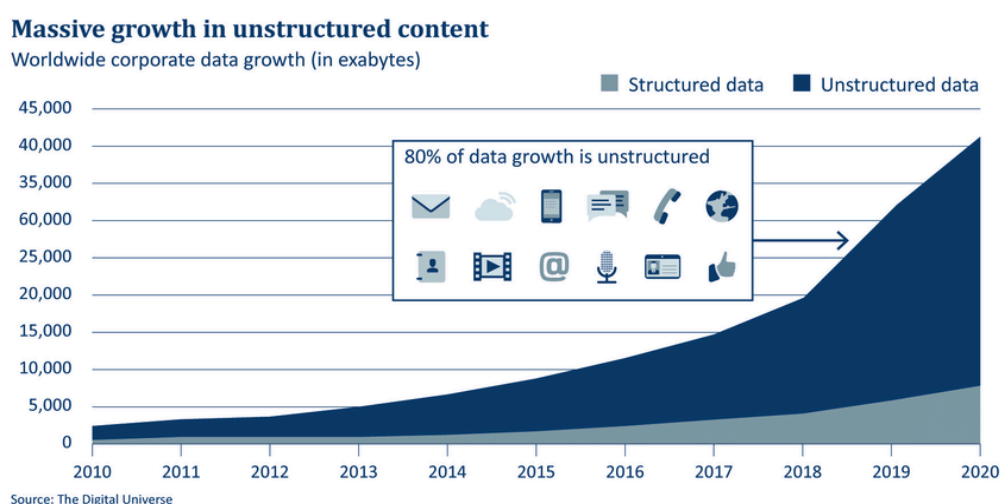
Apesar de ser extremamente importante para certas cargas de trabalho, a computação em nuvem não precisa ser uma solução universal para todos os problemas. Arranjos *multi-clouds* ou *hybrid clouds* estão tornando-se uma alternativa bastante competitiva e uma forma de balancear gastos e minimizar possíveis riscos, enquanto também são capazes de atualizar aplicações legadas e tornar processos mais ágeis.

## 1.2. Armazenamento de Dados

Dados são fundamentais para as mais diversas tarefas computacionais essenciais, sendo que uma grande quantidade de dados precisa ser armazenada (*cold and hot data*) e uma parte desses dados também precisa ser constantemente acessada e rapidamente recuperada (*hot data*) [3].

Uma grande frente de atuação de uma nuvem é a encarregada de armazenar e gerenciar os dados, sejam eles não estruturados (BLOB - *Binary Large Object*), como fotos, áudios, vídeos, textos, ou estruturados, como bancos de dados. Dados não estruturados são chamados de objetos e compõem mais de 80% de toda a “esfera de dados” (*datasphere*) [4].

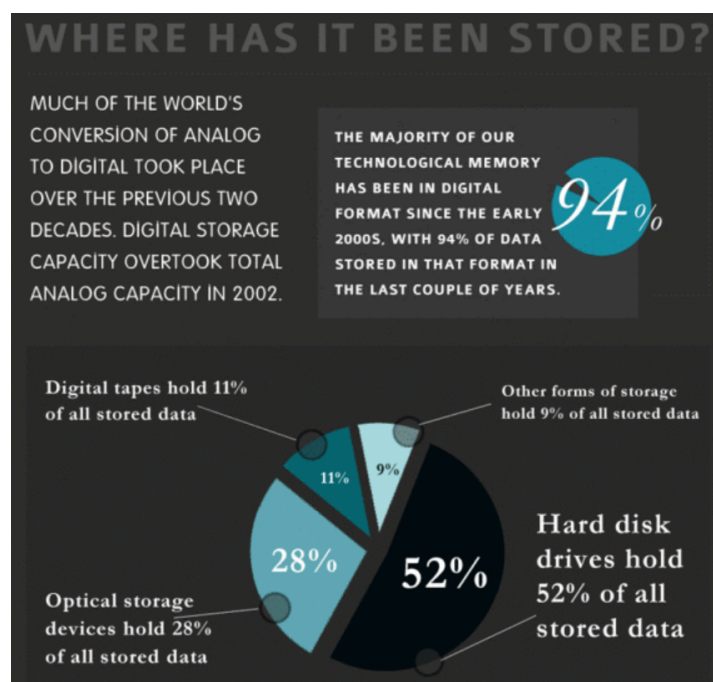
**Figura 1:** Crescimento de dados corporativos não-estruturados mundialmente



**Fonte:** Enabling Big Data Applications for Security do HSG - Hague Security Delta 2017

De acordo com R. J. T. Morris e B. J. Truskowski em "*The evolution of storage systems*" [44], o armazenamento digital já apresentava um melhor custo-benefício do que o uso de papel. Desde então, o custo do armazenamento digital só diminuiu e tornou-se a alternativa mais interessante. Em 2002, foi estimado que 90% de toda nova informação produzida foi armazenada em mídia magnética, discos rígidos em grande parte [5].

**Figura 2:** Distribuição dos dados em diferentes tipos de mídia digital nos anos 2000



**Fonte:** Peter Lyman and Hal R. Varian, 2003.

Como descrito pela Cisco [26], estamos, nos dias de hoje, no que é conhecido como a Era do Zettabyte, que trata-se de um período de tempo que começou em 2010 em que o tráfego na Internet ultrapassou a marca de zettabyte. Um zettabyte é 2 elevado a 70ª potência de bytes, também expressado como  $10^{21}$  bytes, ou seja, 1 bilhão de terabytes ou 1 trilhão gigabytes.

Considerando o que foi discutido até agora sobre o panorama dos dados armazenados em todo mundo, torna-se claro que os dois pontos que têm potencial de apresentar maior impacto no armazenamento de dados em nuvem para as empresas e, conseqüentemente, para os usuários são o estudo dos dados não estruturados e o armazenamento nos discos rígidos.

Além do suporte a grandes volumes de dados, o armazenamento em nuvem tem grandes desafios na escalabilidade dos acessos e na manutenção da consistência das informações armazenadas. Diferentes técnicas de redundância são então usadas tanto para

ampliar a escalabilidade no atendimento a requisições quanto para tentar prover tolerância a falhas.

Por outro lado, lidar com múltiplos dispositivos de armazenamento também significa ter que tratar maiores probabilidades de falhas distribuídas, que devem ser identificadas para ativar o uso das réplicas e também monitoradas para que dispositivos defeituosos sejam substituídos.

Mais do que atuar de forma reativa, contudo, considera-se que o monitoramento contínuo dos dispositivos de armazenamento possa permitir a identificação prévia de situações que indiquem falhas iminentes previsíveis, normalmente devido a deterioração e tempo de uso. Como consequência, a atuação preventiva e até mesmo preditiva no controle de uma infraestrutura de armazenamento de dados em nuvem é comumente realizada através da medição de diferentes parâmetros das operações dos discos. Determinar quais parâmetros monitorar e como usar essas informações na manutenção de uma infraestrutura de computação para armazenamento em nuvem ainda é um desafio.

Neste trabalho, o foco é direcionado ao estudo das métricas *Self-Monitoring, Analysis, and Reporting Technology* (SMART) [6] de HDDs e no estudo de modelos de Aprendizado de Máquina (AM) para predição de falhas em Discos Rígidos para *data centers*. Porém, também foi feito um estudo extenso sobre tecnologias responsáveis pelo armazenamento de objetos (dados não estruturados) em um ambiente de *data center* e nuvem. Mais especificamente, trata-se de estratégias de redundância, do estudo de confiabilidade do sistema, do monitoramento e da manutenção preditiva nos discos.

Como dito por Michael Dell, “*Most customers are beginning to realize that the cloud is not a place, but rather a way of doing IT, where everything is software-defined and infrastructure is code*” [2]. Por isso, esse trabalho trata da computação em nuvem como um meio de realizar computação e como esse meio é construído, com foco na tarefa de armazenamento em *data centers*. Mais especificamente, trata-se de dados armazenados em discos rígidos (HDDs) em muitas máquinas, conseqüentemente aumentando a escala, segurança e flexibilidade dos dados armazenados.

A partir do próximo capítulo, é feito um estudo do funcionamento dos discos rígidos, como os HDs são organizados em um *data center* e um estudo da confiabilidade dos discos. Em seguida, é feita uma amostra do dataset fornecido pela Backblaze e um estudo e explicação das métricas SMART. Após isso, a predição de falha em discos é abordada com o auxílio da manutenção preditiva e aprendizado de máquina.

De forma final, algumas discussões que foram tratadas ao longo deste trabalho são aplicadas de forma mais prática nos últimos capítulos deste documento, enquanto busca-se explicar de forma detalhada as principais etapas. Então, são mostrados os resultados obtidos, que corroboram o que foi tratado no artigo "*Interpretable predictive maintenance for hard drives*" [1], produzindo matrizes de correlação com as métricas SMART mais relevantes, treinando um modelo satisfatório de Árvore de Decisão, com o benefício de ser interpretável, e gerando resultados bons de acurácia, precisão e MCC. Métricas essas que também são comparadas com outros dois modelos comumente aplicados para a predição de falha.

## Capítulo 2. Discos e o Cenário de Armazenamento em Data Centers

Este trabalho tem como grande base de referência os aprendizados, as informações e os dados disponibilizados pelo serviço de armazenamento de dados na nuvem, que é a Backblaze [7]. Essa empresa torna público seus dados e ideias relacionadas ao monitoramento de discos rígidos em seus *data centers* como forma de identificação de condições relacionadas a falhas em HDs.

### 2.1. Funcionamento dos HDDs e uso em Data Centers

Cybersecurity Ventures prevê que a quantidade de dados armazenados na nuvem irá chegar em 100 zettabytes até 2025, ou 50% da quantidade de dados global naquele momento [8]. Considerando toda essa quantidade massiva de dados, os discos usados em *data centers* precisam apresentar grande capacidade de armazenamento e precisam ser usados em grande escala com alta eficiência operacional. O principal fator que gera esse alto nível de eficiência operacional começa com a escolha dos melhores modelos de discos, apesar de também existirem fatores como a eficiência e custo de outras partes da infraestrutura, como os *Storage Pods* que são tratados brevemente ao fim deste capítulo.

O termo “disco rígido” (*Hard Disk* - HD) é na verdade uma abreviação de “unidade de disco rígido” (*Hard Disk Drive* - HDD) e ambos referem-se às mesmas tecnologias de armazenamento. Os discos são incrivelmente duráveis para a intensidade com que são usados em ambientes de alta demanda computacional como os *data centers*. As grandes forças presentes dentro de um disco rígido, suas altas velocidades de giro e a significativa quantidade de engenharia e precisão para gravar e ler corretamente os dados nos pratos são características que fazem dessa tecnologia tão importante e universal. Discos rígidos são a forma mais eficiente para guardar grandes quantidades de dados pelo menor preço, menor \$/Gb.

Atualmente, nos *data centers* da Backblaze [7], por exemplo, os tamanhos dos HDDs normalmente começam com 4TB, tendo sido este o mais utilizado por vários anos até 2018. Naquele ano, já houve um forte uso de discos com 12TB e início do uso de alguns discos ainda maiores, como o de 14TB. Esses tamanhos esses só se tornaram possíveis com o uso cada vez maior de tecnologias como HDs selados e preenchidos com gás hélio que são capazes de diminuir a resistência do ar e economizando a energia em cerca de 20% [9]. Em 2020, é possível notar um forte uso de HDDs de 16TB que tornaram-se prevacentes nos *data centers*. Um maior estudo dos HDs usados e dos dados coletados com as métricas SMART será tratado nas próximas seções, tomando como base o caso dos *data centers* da Backblaze.

Os discos rígidos podem e vão falhar; por esse motivo, sistemas construídos com esses discos devem ser construídos de forma no mínimo resiliente, em que falhas catastróficas, que envolvem perda de dados, serão quase impossíveis de ocorrer. Mas, idealmente, o sistema deve ser quase antifrágil [10], conceito que acredita na existência de um nível acima de robusto e resiliente para algo ser, de tal forma que a falha seja esperada, o sistema trate-a e ainda assim mantenha a mais alta disponibilidade, durabilidade e confiabilidade todo o tempo.

Inicialmente, pensa-se em estratégias para prover maior redundância no sistema de armazenamento, seja no nível da infraestrutura que o *data center* tem, com fontes múltiplas de energia para alimentar os servidores e os sistemas de refrigeração, mesmo em caso de quedas de energia, com o uso de geradores, por exemplo. No nível físico, é comum que busque-se redundância com o uso dos discos em racks, salas, setores, *data centers* ou até zonas de disponibilidade totalmente diferentes.

Dessa forma, os problemas que podem vir a ocorrer em um local muito dificilmente irão afetar outro local físico. No nível lógico dos dados, que pode incluir replicações das informações em mais de um discos em diferentes zonas de disponibilidade e conseqüentemente em diferentes zonas de falha, a chance de mais de um discos que contêm o mesmo dado falhar em um curto período de tempo muito baixa, próxima de 0.

Essas estratégias apresentam um *overhead* significativo mas muitas vezes trata-se de algo necessário para os data centers e para uma plataforma de computação em nuvem. Isso pois infraestruturas de computação em nuvem têm como princípio base prover serviços computacionais que apresentam (1) escalabilidade flexível (potencialmente infinita), (2) custos proporcionais ao uso do usuário, (3) alta disponibilidade, confiabilidade e durabilidade dos dados e serviços.

Com objetivo de chegar mais próximo de atingir essas metas, no nível lógico, existe a redundância dos dados obtida pela replicação, que pode ocorrer com diferentes valores (fator de replicação). Esse número indica a quantidade de vezes que um dado está escrito no sistema de armazenamento, sendo que o fator de replicação 2x apresenta 2 cópias do dado (também entendido como tolerância a 1 falha); conseqüentemente, o fator de replicação 3x apresenta 3 cópias do dado (tolerando 2 falhas) e, assim, sucessivamente. Nessa forma de redundância, cópias completas dos dados são criadas e, com isso, o fator de replicação 2x diminui a capacidade efetiva de armazenamento para 50%; já o fator de replicação 3x diminui a capacidade efetiva de armazenamento para 33%.

Uma estratégia muito usada para reduzir essa significativa sobrecarga (*overhead*) no nível lógico sem que haja perda da confiabilidade do sistema e durabilidade dos dados é o

uso do método conhecido como *Erasure Coding* [11]. Esta técnica necessita de um menor espaço extra para armazenar os dados em mais de uma localidade, pois a replicação não ocorre em todo o dado ou objeto e sim em algumas frações. As técnicas de *erasure coding* dividem os arquivos de dados em  $k$  blocos de dados e  $m$  blocos de paridade e os codifica para que os dados primários sejam recuperáveis sem perda, mesmo que qualquer combinação de  $m$  partes dos fragmentos de dados codificados não estão disponíveis.

Deste modo, por meio de esquemas adequados, que definem o número de partes em que os dados serão fragmentados, a capacidade efetiva de armazenamento consegue ser igual a replicação 2x com 50% de *overhead* com  $k + m = 4+2$ , ou seja, com o mesmo *overhead* mas com maior tolerância a falhas, podendo o dado estar em até 3 dos nós que falhem com  $k + m = 6+3$ . Essa estratégia também tem esquemas capazes de melhorar o *overhead* mínimo, sendo equivalente a uma replicação de fator 1.5x, com 66% de capacidade efetiva de armazenamento (com esquemas 6+2 ou 9+3, por exemplo). Em casos extremos de *erasure coding*, é possível chegar em capacidades reais de até 83%, com fatores na faixa de 1.2x, porém isso necessita de esquemas com pouca tolerância a falha e maior gasto computacional, por exemplo com 6+1. Tudo isso ocorre em troca de um processamento e conseqüentemente tempo extra para reconstruir as partes faltantes a partir de no mínimo  $k$  blocos existentes.

Muitas vezes, ambas as estratégias são utilizadas em um mesmo sistema de armazenamento de dados na nuvem, em que para objetos e conjuntos de dados pequenos, ou para aqueles que são acessados mais vezes, prioriza-se a replicação, por ser menos custosa computacionalmente, mas mais rápida e mais simples. Enquanto isso, *erasure coding* é utilizado na grande maioria dos dados não estruturados [11], pois são dados que apresentam baixa taxa de acesso e, em grande parte dos casos, há grandes volumes de dados.

Na parte operacional e física, também existem alternativas que buscam melhorar o serviço provido, com foco em aumentar a disponibilidade, a durabilidade e a confiabilidade dos HDDs e, conseqüentemente, dos dados armazenados neles. Com o objetivo de entender como é feito em um *data center* que armazena dados para um serviço de nuvem, o exemplo de organização da Backblaze é brevemente tratado a seguir.

### 2.1.1 Como o armazenamento a nuvem da Backblaze é organizado

Tendo em vista a investigação do funcionamento de discos rígidos para a elaboração de fórmulas e modelos de falhas, é relevante entender como é a plataforma de armazenamento

da empresa Backblaze [18]:

- **Drive:** unidade básica de armazenamento, que varia de 4 TB a 22 TB.
- **Storage Server/Pod:** coleção de drives/discos em um único servidor. Existem servidores com 26, 45 e 60 discos, sendo que todos os discos em um servidor de armazenamento têm o mesmo tamanho lógico.
- **Backblaze Vault:** é uma coleção lógica de 20 pods ou servidores de armazenamento, todos com o mesmo número de discos.
- **Tome:** é uma coleção lógica de 20 discos, sendo que cada disco está em um dos 20 servidores de armazenamento em um determinado Vault. Se os servidores de armazenamento em um Vault tiverem 60 discos cada, haverá 60 tomes exclusivos nesse Vault.
- **Cluster:** é uma coleção lógica de Vaults, agrupados para compartilhar outros recursos, como equipamentos de rede e servidores utilitários.

Com base nisso, por exemplo, um *Vault* composto por 20 servidores de armazenamento de 60 discos terá 1.200 discos, um *Vault* com servidores de armazenamento de 45 discos terá 900 discos e um *Vault* com servidores de 26 discos terá 520 discos. Um *cluster* pode ter qualquer combinação de tamanhos de *Vault*.

Um *Backblaze Storage Pod* é apenas uma parte da construção de um serviço de armazenamento em nuvem. Para inicializar a máquina, é necessário ter uma camada de software. A Backblaze usa apenas os softwares livres Debian e Apache para conectar os pods à rede. Começando pela parte inferior, existem 45 discos rígidos ligados através de controladores SATA. Em seguida, a ferramenta *fdisk* no Linux é usada para criar uma partição por disco.

Neste ponto, um Pod de Armazenamento pode ter uma de duas personalidades: um Pod de Armazenamento individual ou um Pod de Armazenamento Backblaze Vault. Para os Pods de Armazenamento individuais, 15 discos rígidos são agrupados num único volume RAID 6 com dois discos de paridade (dos 15). A matriz RAID é criada com o utilitário *mdadm*.

Para os pods de armazenamento do Backblaze Vault, cada um é um dos 20 pods necessários para criar um Backblaze Vault. Um Backblaze Vault divide um arquivo em 20 partes (17 de dados e 3 de paridade) e coloca uma parte do arquivo em cada um dos 20 pods de armazenamento no Vault. Uma implementação própria do algoritmo Reed-Solomon é utilizada para codificar e distribuir os arquivos entre os 20 pods, alcançando 99,99999% de durabilidade dos dados. Essa implementação da codificação Reed-Solomon está disponível

como código aberto em <https://github.com/Backblaze/JavaReedSolomon>.

Para ambos os tipos de *Pods*, de armazenamento e do Vault, o sistema de arquivos EXT4 é usado e o acesso a esses sistemas só é totalmente permitido de forma independente por meio de HTTPS executando a lógica da camada de aplicativo Backblaze personalizada no Apache Tomcat. Um dos conceitos mais importantes aqui é que para armazenar ou recuperar dados com um Backblaze Storage Pod ou um Backblaze Vault Pod, isso deve sempre ocorrer através de HTTPS. Não há iSCSI, NFS, SQL ou Fibre Channel (FCo). Nenhuma dessas tecnologias é escalonada de forma tão barata e confiável nem pode ser gerenciada tão facilmente quanto os pods com seu próprio endereço IP aguardando solicitações em HTTPS. Foi também criada uma própria camada de software para monitorizar pods e vaults, decidir onde armazenar os dados e como encriptá-los, deduplicá-los e indexá-los.

A utilização de modelos diversos para prever falhas nos discos é uma boa maneira de melhorar as garantias necessárias para os HDDs e, conseqüentemente, para os dados armazenados neles. Essa abordagem, que busca tornar o sistema mais robusto e preditivo à falhas, é tratada por alguns estudos que consideram o uso de técnicas de IA para a previsão de falhas em HDs. Isso ocorre pois o problema da análise de métricas SMART e de vida útil do disco é complexo mas, ainda assim, ao envolver grandes quantidades de dados é possível chegar em modelos com resultados bastante satisfatórios. Neste trabalho, o foco do estudo e experimentação está no uso de Modelos de Aprendizado de Máquina [12], principalmente das Árvore de Decisão (*Decision Trees* ou DTs), para prever a falha em um modelo específico de disco.

## 2.2. Métricas de Confiabilidade

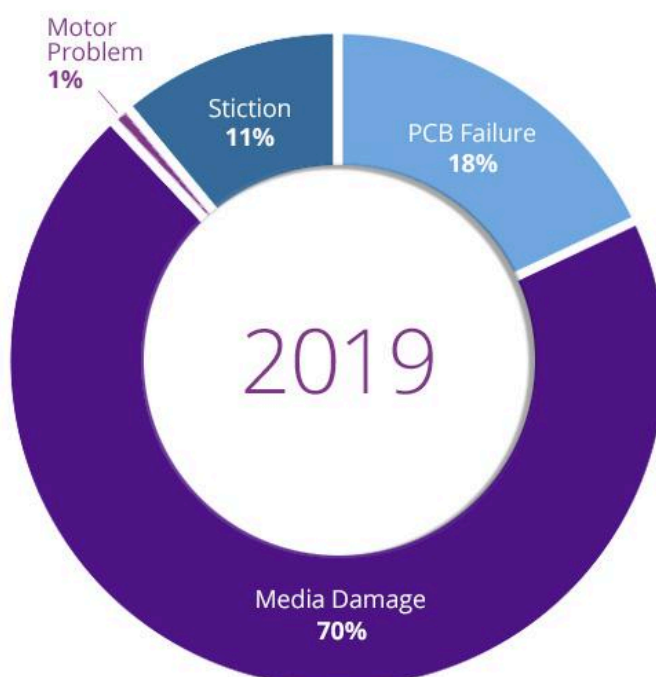
Grandes empresas de nuvem são capazes de armazenar dados de seus clientes com uma promessa de disponibilidade e durabilidade realisticamente perfeita. Isso significa prover um valor de durabilidade na ordem de 9 ou mais noventa e disponibilidade altíssima, com o serviço não estando funcional por apenas alguns segundos no ano [13]. Esses valores parecem ser difíceis de serem atingidos; no entanto, quando se pensa em técnicas de redundância, como as descritas no capítulo anterior, de Replicação e de *Erasure Coding*, a probabilidade de todos os discos que armazenam um objeto falharem em um período curto que é MTTR (*mean time to repair* - tempo médio de recuperação da falha) torna-se extremamente baixa. Esse tempo refere-se à identificação da ocorrência de um erro, todo o tempo para a manutenção ocorrer e possível troca de disco, passagem dos dados do disco que falhou para o novo ou de forma distribuída para os discos em uma espécie de Hash

Consistente ou Ring, por exemplo.

Para melhor entender os assuntos tratados neste trabalho, é importante conhecer os principais conceitos relacionados ao tema de confiabilidade (*Reliability Engineering*). Essa área de estudo tem como base o conceito de confiabilidade que representa “qualidade ao longo do tempo” de um sistema ou de um produto. Ou seja, trata-se de uma probabilidade de um produto funcionar corretamente dada uma condição operacional específica em um período de tempo. Trazendo para o caso do armazenamento, a confiabilidade poderia ser uma medida que verifica se o HD está realizando suas operações de leitura e escrita de forma adequada, sem perder ou corromper dados.

Antes de apresentar qualquer conceito que servirá de medida do que acontece com os discos é interessante entender o que compõe uma falha. Em um dos blogs da Backblaze [14], foi apresentada uma análise dos dados obtidos dos discos que chegaram no laboratório de recuperação de dados (*drive savers*), gerando os resultados estão apresentados na Figura 3.

**Figura 3:** Drive Failures by Possible External Force



Fonte: [Backblaze](#), 2019

Alguns dos principais indicadores do funcionamento de discos são Taxa de Falha (*Failure Rate*), Tempo Médio Para Falha (*Mean Time To Fail* - MTTF), Tempo Médio Entre Falhas (*Mean Time Between Failures* - MTBF).

Todos os três são usados em casos diferentes. MTTF normalmente é adequado para produtos irreparáveis, que após sofrerem uma falha são comumente substituídos pois requerem conserto que normalmente não vale a pena financeiramente. MTBF já trata de produtos reparáveis ou de um sistema com múltiplos produtos de uma categoria, como é o caso de um data center com diversos discos rígidos compondo a infraestrutura de armazenamento. A taxa de falha pode ser obtida pelo inverso de uma das medidas anteriores, como mostrado a seguir.

- *Failure Rate* ( $\lambda$ ) é o conceito que representa a taxa (número de vezes) que um produto falha em um certo período de tempo. Pode ser calculado pelo Número de Falhas dividido pelo Tempo Operacional (Ciclos), normalmente em horas, e o resultado é em Falhas por Hora ou, no caso de HDDs, comumente é anual.

$$\text{Failure Rate } (\lambda) = \frac{\text{Number of Failures}}{\text{Operating Time (Cycles)}} = \text{Failures Per Hour } (1)$$

- MTTF e MTBF ( $\theta$ ) são o inverso da medida anterior e a única diferença dessas duas medidas é que a primeira trata de produtos com falhas que não são reparáveis, por isso é o tempo até a falha. Enquanto o MTBF trata de produtos reparáveis (ou irreparáveis mas substituíveis no sistema) uma vez que apresentar uma falha, por isso é o tempo entre falhas.
- Ambos representam o tempo médio para ocorrer uma falha

$$\text{MTTF} = \text{MTBF} = \theta = \frac{\text{Operating Time (Cycle)}}{\text{Number of Failures}}, (2)$$

dessa forma,  $\lambda = 1/\theta$ . (3)

Uma boa alternativa para entender esses dados de forma mais didática, mas não exata, é pensar que um valor de MTBF de 1 milhão de horas, ou seja 114 anos, indica que caso haja 114 discos, espera-se que 1 falhe por ano em média. O que torna essa afirmação não exata é o fato de que essa estimativa é um estimativa média, sem levar em conta diversas complexidades do modelo e especificidades do sistema de armazenamento que utiliza dos discos.

Sistemas de armazenamento que, assim como todos os outros, têm um tempo esperado de funcionamento com base em suas tecnologias e mecanismos utilizados. Dessa forma, sua vida útil pode ser descrita por um modelo que estima diversas variáveis, como MTBF (*Mean Time Between Failures*), MTTR (*Mean Time To Repair*), Disponibilidade, Confiabilidade, Tolerância a Partição.

No entanto, modelos tradicionais, como Markov Chains (uma espécie de máquina de estados que com base na probabilidade de uma medida ocorrer ou não muda de estado) não tratam de casos de cisnes negros (*black swans* e em alguns casos *grey swans*) [15]

Exemplo de cálculo de durabilidade para um sistema de armazenamento:

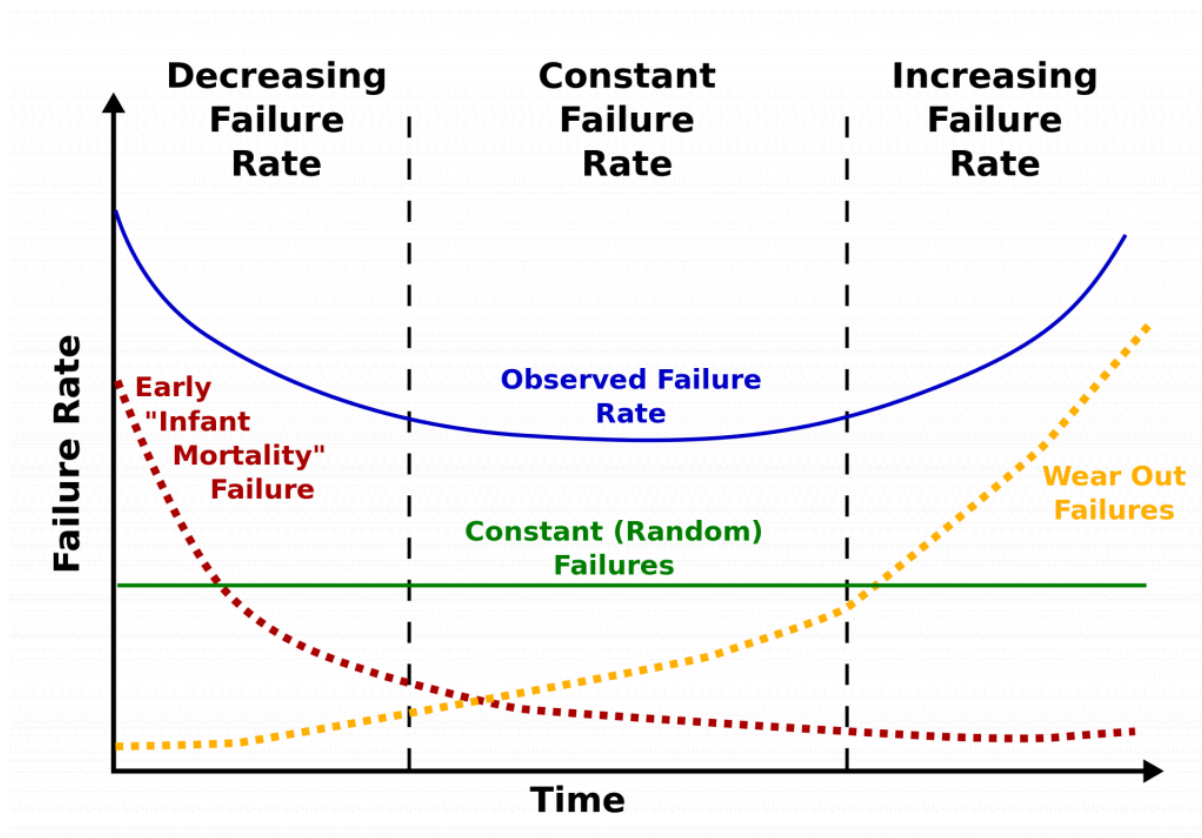
- Supondo um cálculo de durabilidade, procura-se obter o número de 9's serve na verdade como limite superior, sem contabilizar eventos catastróficos. Algumas empresas de nuvem promete 12 9's de durabilidade, mas se aplicar essa fórmula o número de 9's passa de 20 mas esse valor é teórico em que as condições a seguir sejam apenas:
- taxas de falha  $\lambda = (1/MTTF)$
- taxa de recuperação  $\mu = (1/MTTR)$
- n discos que falham, cada um a uma taxa de  $\lambda$  falhas/hora
- um disco com falha, que é recuperado a uma taxa de  $\mu$  recuperações/hora

$$data\ loss = \frac{n!}{m!(n-m-1)!} \times \frac{\lambda^{m+1}}{\mu^m} \text{ per hour (4)}$$

### 2.2.1 Curva da Banheira (*Bathtub Curve*)

Atualmente, Engenheiros de Confiabilidade (SREs) [16] usam uma curva para descrever as taxas de falha esperadas em cada período. Basicamente, trata-se de um modelo da confiabilidade de uma unidade de um produto genérico ao longo do tempo. A principal ideia é que os defeitos vêm de três fatores: (1) defeitos de fábrica, que causam boa parte da alta taxa de falha nos primeiros momentos (*“infant mortality”*); (2) falhas aleatórias, aproximadamente constantes; e (3) partes que se desgastam naturalmente, após muita utilização. O gráfico apresentado na Figura 4 mostra como esses três fatores podem produzir a curva de taxa de falha no formato de uma banheira.

Figura 4: Bathtub Curve



Fonte: [https://en.wikipedia.org/wiki/Bathtub\\_curve](https://en.wikipedia.org/wiki/Bathtub_curve). Acesso em 20/09/2024

No período intermediário, também chamado de período de vida útil do produto, a taxa de falha é aproximadamente constante e, por isso, é possível utilizar a distribuição exponencial para fazer predições da confiabilidade do produto.

$$R(t) = e^{-\lambda t} \quad (5)$$

De forma mais precisa, é possível descrever a curva da banheira com a utilização da distribuição de *Weibull* para cada um dos 3 principais períodos na vida do produto.

$$R(t) = e^{-\left(\frac{t}{\theta}\right)^\beta}, \quad (6)$$

em que:

$\beta$  - parâmetro de forma de Weibull

$\theta$ , MTTF ou MTBF - parâmetro de escala de Weibull

No primeiro período, a curva apresenta um parâmetro  $\beta < 1$  (taxa de falha  $\lambda$

decrecente), depois  $\beta = 1$  ( $\lambda$  constante) e por fim  $\beta > 1$  ( $\lambda$  crescente). Curiosamente,  $\beta = 3.5$  aproxima a curva da distribuição normal.

### 2.2.2 Cálculo de AFR (*Annualized Failure Rate*)

A taxa de falha anual (AFR) é uma estimativa da porcentagem de produtos que falham em um ano, em vez de ser uma medida com médias, como é feito no MTBF. Para calcular uma taxa de falha de um determinado grupo de discos durante um determinado período de tempo, são utilizados os dados de número de discos e o número de falhas de discos durante esse período de tempo. O período de tempo em questão é definido como 1 ano, ou seja 8766 horas. Assim, se durante o último ano, de 10 discos, um falhou, pode-se dizer que houve uma taxa de falha de 10% para o ano. *Data centers* são diferentes pois não se tratam de sistemas estáticos, novos discos entram e saem do sistema diariamente e há discos novos, discos com falhas, discos migrados e assim por diante. Em outras palavras, o número de discos não é consistente num determinado período de tempo.

Para resolver esse problema, nasceu a ideia dos **dias de disco** (*Drive Days*), que também entra na fórmula de MTBF como tempo operacional. Um dia de disco corresponde a um disco em funcionamento durante um dia. Portanto, um disco em funcionamento durante dez dias corresponde a dez dias de disco e dez discos funcionando por um dia também correspondem a dez dias de disco.

Para ver isto em ação, é definida a coleção de discos e o período de tempo desejado e, em seguida, basta aplicar a seguinte fórmula para obter a taxa de falha anualizada (AFR) em taxa unitária, bastando multiplicar por 100 para obter a %.

$$AFR = \text{Falhas de disco} / (\text{Dias de disco} / 365), (7)$$

onde:

- *Falhas de disco* é o número de discos que falharam durante o período de observação.
- *Dias de disco* é o número de dias em que todos os discos observados estiveram operacionais durante o período de observação.

Este cálculo simples permite calcular uma Taxa de Falhas Anualizada (*Annual Failure Rate* - AFR) para qualquer conjunto de discos durante qualquer período de tempo e trata um número variável de discos durante esse período. [17]

O termo Taxa de Falhas Anualizada é altamente utilizado, sendo que a palavra “anualizada” aqui significa que, independentemente do período de observação (mês,

trimestre, etc.), a taxa de falhas será transformada numa medida anual. Para um determinado grupo de discos (ou seja, modelo, fabricante, etc.), calcula-se a AFR para um período de observação.

De modo mais específico ao caso que está sendo estudado, por exemplo, Backblaze começou a coletar dados em 10 de abril de 2013, mas muitos dos discos estavam presentes antes disso. Caso seja do interesse calcular o AFR de um modelo X para 2013, não seria possível contar o número de dias de disco que esses discos tiveram antes de 10 de abril. A solução está relacionada ao valor bruto (*raw*) do atributo SMART 9, que contém o número de horas de ativação de um disco. Com isso, é possível obter o número de dias, ou seja, os dias de disco.

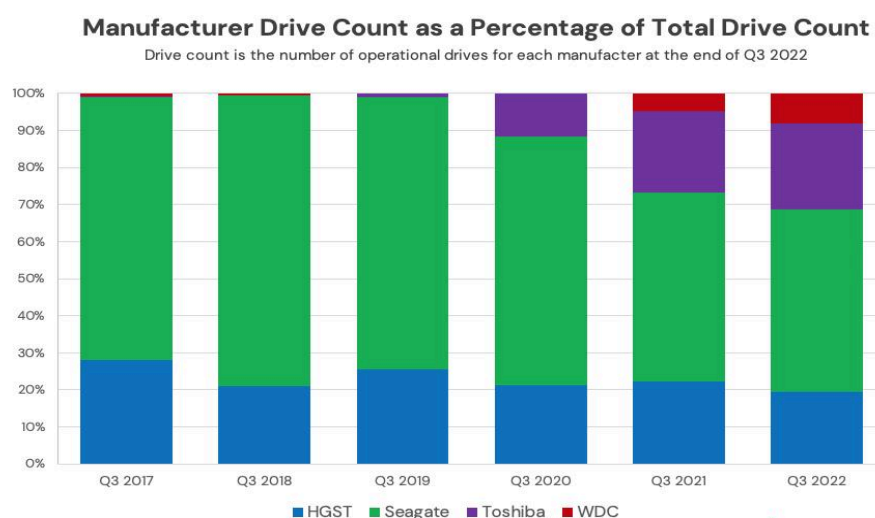
## Capítulo 3. Métricas SMART para HDs e Dataset da Backblaze

As métricas SMART (*Self-Monitoring, Analysis, and Reporting Technology*) tratam-se de um sistema de monitoramento de HDDs e SSD, responsável por relatar indicadores e medidas de aspectos de confiabilidade de discos. Esses dados são comumente usados para prover conhecimento da vida de um disco, suas falhas e possíveis causas. Para melhor entender a origem dos dados SMART que são tratados ao longo do trabalho, a seguir, é feita uma breve discussão dos discos da Backblaze que as métricas foram retiradas.

O ambiente de armazenamento de dados na Backblaze apresentou no segundo trimestre de 2024 uma contagem de discos de 284,386 (HDDs e SSDs), sendo que todos esses dispositivos são monitorados e geram arquivos de registro diariamente. O conteúdo dos arquivos é tratado de forma detalhada na próxima seção.

Os discos rígidos que são usados na plataforma Backblaze são unidades padrão de 3,5 polegadas. Inicialmente, a maior parte desses discos era da categoria para consumidores, que podem ser comprados *on-line* ou em lojas mas, após uma certa escala da nuvem e após 2015, alegam comprar cada vez mais modelos de HDDs empresariais diretamente com fabricantes. A redundância fornecida por softwares também busca garantir uma segurança dos dados e permite usar unidades prontas para uso dos principais fabricantes de unidades de disco, sendo os maiores deles: Seagate, Western Digital e Toshiba. O gráfico na Figura 5 detalha a contagem atual de HDs por fabricante.

**Figura 5:** Percentual de discos por fabricante na nuvem BackBlaze



Fonte: [Backblaze](#), 2022.

### 3.1. Visão geral do Dataset da Backblaze

Ao longo dos anos, o dataset Drive Stats [17] tem sido usado de várias maneiras e por diferentes razões. Utilizando o Google Scholar, é possível encontrar atualmente 660 citações para o termo “*Backblaze hard drive stats*” desde 2014. Isso inclui 18 artigos de revisão. Ao trabalhar com esse dataset, é natural que se entenda mais sobre limpeza e tratamento de dados de informações coletadas de HDs em uso pleno. Também é bastante relevante a abordagem de tentar aplicar técnicas de Inteligência Artificial e Aprendizado de Máquina aplicadas aos dados do Drive Stats para determinar a falha ou a integridade de uma unidade ou até mesmo quanto tempo de vida restante um dispositivo tem.

Os dados do conjunto *Drive Stats* também são usados internamente pelos funcionários da Backblaze para informar sobre as operações e executar experimentações próprias. Alguns exemplos desses usos são listados a seguir.

Parte do processo de desenvolvimento e produção do Backblaze Storage Pod foi o desenvolvimento do software para gerenciar o próprio sistema. Quase desde o primeiro dia de funcionamento dos discos, determinados dados históricos e atuais estatísticas SMART mais relevantes foram coletadas diariamente de cada disco e registrados para serem usados posteriormente em análises e para treinar modelos de predição de falha dos HDDs. Na prática, outros sinais, como erros de ATA ou alertas de FSCKs, também podem fornecer o primeiro indicador de um problema.

Acredita-se fortemente, que os dados SMART estão correlacionados com falha e com os dados sobre *Remaining Useful Life* (RUL) de um disco específico. Com essas informações, torna-se possível saber o momento de substituir um disco.

Os dados fornecidos pela Backblaze dos discos rígidos de seus *data centers* começam em 2013 e acompanham relatórios anuais e trimestrais. No ano de 2023 foram obtidos os seguintes valores:

- Drive Count: 269,756
- Drive Failures: 16,962
- Drive Days: 424,474,539
- Annualized Failure Rate: 1.46%

Esses dados são bastante completos e vem de um ambiente de produção real, de considerável escala, tudo isso disponibilizado de forma gratuita, apesar de extremamente valiosos para pessoas cientistas de dados, para estudantes de IA e para engenheiros de confiabilidade que queiram entender mais sobre armazenamento e seu monitoramento.

Todos os dias, no centro de dados da Backblaze, é feito um registro ou linha de dados que inclui informações básicas de cada disco, juntamente com as estatísticas SMART relatadas por esse disco. Todos os registros do disco para um determinado dia são reunidos em um arquivo que consiste numa linha para cada disco rígido ativo. O formato deste arquivo é ".csv" (*Comma Separated Values*). Cada dia este arquivo é nomeado no formato AAAA-MM-DD.csv, por exemplo, 2013-04-10.csv.

A primeira linha de cada arquivo contém os nomes das colunas, as restantes linhas são os dados reais. As colunas são as seguintes:

- Date – A data do arquivo no formato aaaa-mm-dd.
- Serial Number – O número de série do disco atribuído pelo fabricante.
- Model – O número do modelo do disco atribuído pelo fabricante.
- Capacity – A capacidade do disco em bytes
- Failure – “0” se o disco estiver OK. Contém um “1” se este for o último dia em que a unidade esteve operacional antes de falhar.

Esses arquivos diários .csv gerados com as informações de todos os discos em funcionamento são agrupados em um arquivo que contém todos os dados coletados de 1 trimestre do ano, arquivo esse que é compactado e disponibilizado gratuitamente no site em: <https://www.backblaze.com/cloud-storage/resources/hard-drive-test-data>.

### 3.2. Atributos SMART com maior correlação com falhas em discos

SMART (*Self-Monitoring, Analysis, and Reporting Technology*) [6] é um padrão para monitoramento de discos rígidos que coleta diversas métricas relacionadas ao desempenho e saúde do disco. Como será observado a seguir, diferentes trabalhos e experimentos vão indicar diferentes métricas SMART como maiores variáveis correlacionadas com as falhas nos discos. O objetivo aqui é apanhar o máximo dessas técnicas e apresentar que diferentes discos, em diferentes situações irão ter diferentes atributos mais relevantes para indicarem falhas futuras. Por esse motivo, o uso de abordagens provenientes do Aprendizado de Máquina demonstram bons resultados em comparação a outras estratégias.

**Tabela 1:** Alguns dos Atributos SMART para predição de falha em discos

ID	Attribute Names	ID	Attribute Names
1	Read Error Rate	7	Seek Error Rate
9	Power-On Hours	192	Power-off Retract Count
10	Spin Retry Count	193	Load/Unload Cycle Count
3	Spin-Up Time	194	Temperature
12	Power Cycle Count	197	Current Pending Sector Count
4	Start/Stop Count	198	Uncorrectable Sector Count
5	Reallocated Sectors Count	199	UltraDMA CRC Error Count

**Fonte:** Sidi Lu e Bing Luo, 2020.

As principais métricas SMART úteis para estimar o RUL (*Remaining Useful Life*) de um HDD serão explicadas a seguir:

- **Reallocated Sectors Count - 5** (Setores Realocados): Conta o número de setores defeituosos que foram realocados. Um aumento constante nessa métrica pode indicar degradação iminente.
- **Spin Retry Count - 10** (Contagem de Retentativas de Giro): Mede o número de tentativas adicionais necessárias para girar o disco a uma velocidade operacional. Um aumento nesta métrica pode indicar falha iminente nos componentes mais físicos como no motor ou no mecanismo de giro.
- **Temperature - 194** (Temperatura): Monitora a temperatura do disco. Temperaturas consistentemente elevadas podem causar degradação mais rápida e reduzir o RUL.
- **Power-On Hours - 9** (Horas Ligadas): O número total de horas que o disco esteve operacional. Pode ser um indicativo de desgaste e tempo de vida restante, especialmente quando combinado com outras métricas.
- **Seek Error Rate - 7** (Taxa de Erro de Busca): Monitora o número de erros de busca ao tentar localizar dados no disco. Um aumento nesta métrica pode indicar falhas na cabeça de leitura/gravação.
- **Current Pending Sector Count - 197** (Contagem de Setores Pendentes): Conta os setores problemáticos que não foram lidos corretamente e aguardam para serem remapeados. Um valor elevado pode ser um sinal de falha iminente.

De acordo com dados divulgados de experimentos feitos pela própria Backblaze [21], as métricas SMART cruciais para determinar da melhor forma a saúde ou falha iminente de um disco rígido (HDD) estão listadas aqui.

- SMART 5: Reallocated\_Sector\_Count. [19]
- SMART 187: Reported\_Uncorrectable\_Errors.
- SMART 188: Command\_Timeout.
- SMART 197: Current\_Pending\_Sector\_Count.[20]

- SMART 198: *Offline\_Uncorrectable\_Sector\_Count*. [20]

Por meio das informações presentes em outros estudos e experimentações, a base de dado que foi usada neste trabalho também adicionou os seguintes atributos (marcados como possíveis indicadores de acordo com outros estudos citados na wikipédia):

- SMART 1: Read Error Rate [22]
- SMART 3: Spin-Up Time [22]
- SMART 7: Seek Error Rate [22]
- SMART 10: Spin Retry Count [20]
- SMART 184: End-to-End error / IOEDC [20]
- SMART 190: Temperature Difference
- SMART 196: Reallocation Event Count [20]

No entanto, em uma das primeiras etapas de tratamento de dados, o atributo 196 foi retirado do conjunto de dados deste estudo pois, nos discos da Seagate de 4TB selecionados, esse atributo apresentava apenas valores nulos, o que indica que ele não é monitorado. As demais métricas selecionadas para este estudo podem ser resumidas como segue:

- **SMART 5: *Reallocated Sector Count*** (Contagem de Setores Realocados): Esta métrica contabiliza o número total de setores defeituosos que foram realocados para um local de reserva (ou seja, um "setor de reserva" no disco). Quando o HDD detecta que um setor se tornou ilegível ou instável, ele transfere automaticamente os dados desse setor para um setor de reserva e marca o setor original como "realocado". Um aumento no número de setores realocados geralmente indica uma degradação física do meio de armazenamento. Se essa métrica começa a aumentar, significa que o disco está começando a desenvolver setores defeituosos, o que pode ser um sinal precoce de falha iminente. Frequentemente, um aumento contínuo na contagem de setores realocados precede falhas mais graves, como uma falha catastrófica do disco. O aumento desse contador é um forte sinal de alerta de que o disco está se aproximando do final de sua vida útil.
- **SMART 187: *Reported Uncorrectable Errors*** (Erros Irrecuperáveis Reportados): Essa métrica monitora o número total de erros de leitura que não puderam ser corrigidos pela lógica de correção de erros do disco. Esses erros são considerados "irrecuperáveis" e, geralmente, indicam que os dados no setor afetado foram perdidos ou danificados permanentemente. Um aumento nos erros irrecuperáveis é um indicador claro de falhas de leitura que o disco não consegue corrigir

automaticamente. Isso pode ser causado por danos físicos na superfície do disco, problemas com a cabeça de leitura/gravação, ou problemas elétricos. Caso a contagem de erros irreversíveis reportados aumente, é provável que a falha do HDD seja iminente, pois o disco está falhando em sua função básica de ler e gravar dados de forma confiável.

- **SMART 188: *Command Timeout*** (Tempo de Expiração de Comando): Esta métrica mede o número de vezes que uma operação de comando do disco (como leitura ou gravação) excedeu o tempo limite permitido. Isso pode ocorrer devido a problemas de comunicação entre o disco e o sistema ou devido a falhas internas do disco, como falhas mecânicas ou elétricas. Problemas de Comunicação ou Falhas Internas: Um aumento no valor de Command Timeout pode indicar problemas no controlador do HDD, na interface de comunicação (por exemplo, SATA), ou falhas intermitentes que fazem com que o disco perca ou interrompa o sinal de comando. Se o número de timeouts aumentar consistentemente, isso pode ser um sinal de que o HDD está enfrentando problemas graves que podem levar a uma falha completa. Esses problemas podem incluir falhas no motor de giro, cabeças de leitura/gravação defeituosas, ou problemas de firmware.
- **SMART 197: *Current Pending Sector Count*** (Contagem de Setores Pendentes): Esta métrica monitora o número de setores que estão “pendentes” para serem remapeados, pois o disco rígido encontrou problemas ao tentar lê-los. Se o disco conseguir ler com sucesso esses setores em uma tentativa futura, eles serão remapeados; caso contrário, serão marcados como setores irreversíveis. Um valor elevado nesta métrica indica que o disco está encontrando problemas em vários setores, o que significa uma alta probabilidade de perda de dados por conta de dados corrompidos. À medida que a contagem de setores pendentes aumenta, é provável que o disco esteja em um estado de degradação contínua e progressiva. Muitos setores pendentes geralmente precedem um aumento na contagem de setores realocados ou erros irreversíveis, ambos indicadores críticos de falha iminente.
- **SMART 198: *Offline Uncorrectable*** (Erros Offline Irreversíveis). Este contador rastreia o número de erros que não puderam ser corrigidos durante verificações "offline" de integridade, que são rotinas de varredura do disco que acontecem quando o disco não está em uso ativo pelo sistema. Esses erros são identificados quando o disco tenta realizar operações de manutenção ou verificação enquanto

está ocioso. Um aumento nos erros offline irrecuperáveis geralmente indica danos substanciais na superfície do disco ou falhas mecânicas. Esses erros sugerem que partes do disco estão tão danificadas que o próprio disco não pode mais fazer leituras ou correções bem-sucedidas. Essa métrica é um dos sinais mais claros de uma falha de disco iminente. Um valor crescente de erros offline irrecuperáveis indica que o disco provavelmente já perdeu a capacidade de funcionar corretamente e que uma falha completa pode ocorrer a qualquer momento.

Por fim, as métricas 1, 3, 7, 10, 184 e 190 podem ser estudadas mais a fundo nas referências citadas anteriormente, seja pelo documento de especificação de atributos SMART da Seagate (para 1,3 e 7), pelo portal de Suporte da Acronis (para 184 e 196), ou por outros estudos no caso da diferença de temperatura (190). Algumas dessas medidas têm seus significados e implicações abordados neste trabalho.

### 3.3. Dados Empíricos obtidos da Backblaze

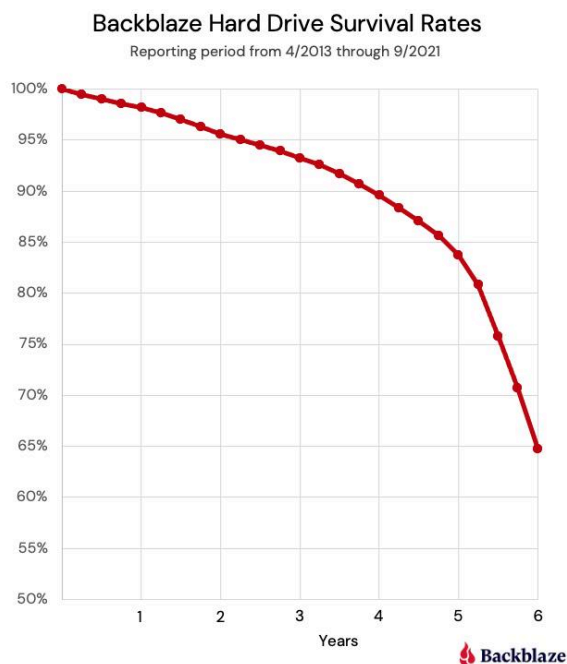
Backblaze usa atualmente mais de 200.000 discos rígidos para armazenar os dados dos clientes [23]. Os discos variam de 4TB a 18TB de tamanho. Quando somados, existem mais de dois exabytes de espaço em discos rígidos sob gerenciamento. A maioria destas unidades está montada num servidor de armazenamento que acomoda 60 unidades, mais um disco para boot. Muitos servidores de armazenamento utilizam apenas 45 discos rígidos, como mostram dados de dezembro de 2021. Os servidores de armazenamento consistem em Storage Pods e servidores de armazenamento de fabricantes externos. Vinte servidores de armazenamento são agrupados em um Backblaze Vault, que utiliza um algoritmo de codificação de eliminação Reed-Solomon para replicar e armazenar dados de clientes nos 20 servidores em um Backblaze Vault.

De acordo com os dados, os discos da Seagate tendem a apresentar taxas de falha mais altas, com alguns modelos de 12-14 TB atingindo uma AFR de 4-5%. Por outro lado, os discos da Seagate também são mais baratos e a Backblaze ainda vê o valor de usar discos da Seagate para sua operação em grande escala. No entanto, talvez não tanto se a necessidade envolver um ou dois discos altamente confiáveis. Para uma operação de armazenamento em nuvem, nenhum dos discos nos dados mostra problemas sérios - mesmo aqueles com o AFR mais alto ainda são viáveis na escala da Backblaze. Caso seja de interesse identificar a marca mais confiável, de acordo com os dados coletados e disponibilizados na plataforma, a Western Digital tem os melhores números até o momento (12/2021). A HGST, uma submarca da Western Digital agora descontinuada, também

apresenta uma confiabilidade sólida. A Toshiba é a segunda mais confiável, seguida da Seagate. Vale a pena mencionar que mesmo os piores AFRs ainda são algo em torno de alguns por cento após anos de serviço.

Com menos de 36 meses, nenhum disco tem mais de 1,5% de AFR. Após 36 meses, não há uma grande quantidade de dados para outros fabricantes, mas os discos da Seagate são os que ultrapassam o limite de 1,5%. [24]

Em relação às taxas de sobrevivência dos discos, é surpreendentemente difícil obter uma resposta à pergunta “Quanto tempo dura um disco rígido?”. Para isso, foi utilizada a ajuda de um dos posts da equipe da Backblaze, que pode ser encontrado aqui. Embora os relatórios de AFR dos discos respondam à pergunta sobre a taxa de falhas dos discos, não dizem muito sobre a sua duração. Curiosamente, os mesmos dados recolhidos e utilizados para prever a falha de discos podem ser utilizados para calcular a expectativa de vida dos modelos de discos rígidos que são utilizados. É tudo uma questão de como se olha para os dados. Quando as técnicas de previsão da expectativa de vida são aplicadas aos dados de discos que foram recolhidos, os dados da Backblaze produzem os resultados apresentados na Figura 7.

**Figura 7:** Taxa de Sobrevivência dos HDs da Backblaze

Fonte: [Backblaze](#), 2024.

De acordo com os dados monitorados pela Backblaze [23], a expectativa de vida de discos rígidos diminui a uma taxa bastante estável de 2% a 2,5% por ano durante os primeiros quatro anos, depois disso a diminuição começa a acelerar. Olhando para o gráfico AFR por trimestre acima na figura 7, isto faz sentido, uma vez que a taxa de insucesso aumenta a partir do quarto ano. Ao fim de seis anos, vê-se uma expectativa de vida de 65%. Dito de outra forma, se um disco rígido foi comprado há seis anos, há 65% de chance de ainda estar vivo hoje.

De uma forma mais específica, todos os discos de 4 TB têm taxas de falha aceitáveis, mas foram comprados principalmente os discos da Seagate. A disponibilidade imediata e o preço altamente competitivo dos discos de 4 TB da Seagate, juntamente com seu desempenho sólido e taxas de falha respeitáveis, fizeram deles a melhor escolha.

Uma observação relevante da equipe de operações da backblaze sobre os discos da Seagate é que eles geralmente sinalizam a falha iminente através das estatísticas SMART. Como várias estatísticas SMART são monitoradas, muitas vezes problemas são identificados antes de uma falha pendente e torna-se possível tomar as medidas adequadas. As falhas de discos de outros fabricantes parecem ser menos previsíveis por meio das estatísticas SMART.

Considerando essas observações, os discos de 4TB da Seagate foram considerados os candidatos mais apropriados para serem usados em um estudo que busca entender

como criar e aplicar modelos de predição de falhas para esses discos. Dessa forma, busca-se, idealmente, identificar uma forma confiável de fazer uma manutenção preventiva, com o objetivo de controlar e impedir falhas a partir do monitoramento das partes de um sistema que utilize múltiplos discos para o armazenamento em nuvem.

## Capítulo 4. Predição com Modelos de AM e Trabalhos Relacionados

Neste Capítulo, é feito um apanhado de conhecimentos e conceitos relacionados à capacidade preditiva de modelos de Aprendizado de Máquina, juntamente com citações de trabalhos relacionados ao foco deste trabalho, unindo teoria e prática computacional. A grande área em questão é a de Manutenção Preditiva, que busca definir se dispositivos ou sistemas estão prestes a apresentar falha ou não, ou quanto tempo estima-se haver até uma falha e possíveis otimizações que podem ser realizadas com esses conhecimentos.

Os principais casos de uso de modelos de AM em manutenção preditiva são:

- Para prever se uma falha vai ou não ocorrer com determinados produtos (problema de Classificação Binária) em que os dados que se tem acesso é de dados censurados
- Para estimar a Vida Útil Restante (RUL) de aparelhos, como HDs. Muitas das vezes, trata-se de um cenário que utiliza técnicas de regressão.
- Um caso mais avançado e menos usado é o de otimização de cenário. Para esse caso, é necessário que já exista um modelo implementado de forma madura para aplicar maneiras de tornar a solução mais otimizada por outros algoritmos [25].

Em grande parte, os modelos de Aprendizado de Máquina considerados neste trabalho e em trabalhos relacionados consistem em Modelos Supervisionados, pois o aprendizado ocorre a partir de dados já rotulados e as classes já são conhecidas. O foco dos modelos que serão construídos é aprender as relações complexas entre as métricas SMART mais relevantes as falhas nos discos, de forma a ser capaz de generalizar para novas amostras. A classificação binária do estado do HDD, de apresentar falha iminente em uma janela de tempo ou não, é a principal tarefa da manutenção preditiva investigada neste trabalho.

**Tabela 2** - Distribuição de contagem de falhas do tempo de vida

	# of Failures	% of Drives	% of Failed Drives
SSD	0	88.71	—
	1	10.10	89.60
	2	1.038	9.208
	3	0.133	1.180
	4	0.001	0.001
HDD	0	93.35	—
	1	6.64	99.81
	2	0.01	0.194

**Fonte:** Riccardo Pinciroli, 2022. [30]

Na Tabela 2, é possível notar a alta correlação entre o atributo *failure* e falha propriamente dita no HDD. Com isso, os eventos de falha podem ser diretamente registrados pelo atributo em questão. Para SSDs, isso já não é tão simples e direto pois, nas observações realizadas pela Backblaze [7], 9.2% dos SSDs que apresentaram um atributo de falha “*failure*” 1 vez não falharam de forma total e apenas na segunda vez o total de discos que falharam chegou em 98.8%. Em alguns (1.18%) dos discos que falharam, a falha só se concretizou na terceira vez que o atributo “*failure*” apareceu.

Na manutenção preditiva de HDDs, geralmente tem-se as métricas SMART como componentes de grande importância, uma vez que podem identificar precocemente os sinais de degradação e apoiar a tomada de ações preventivas (como *backups* ou substituições) antes que ocorram falhas catastróficas. Além disso, métricas como as mencionadas são frequentemente usadas em modelos para prever a vida útil remanescente (RUL) do disco, ajudando a otimizar a substituição e manutenção dos equipamentos. As três principais formas de estimar RUL de equipamentos serão tratadas a seguir.

#### 4.1. Manutenção Preditiva

De acordo com Nassim Nicholas Taleb, autor do livro *The Black Swan* [15], que trata de eventos improváveis e como eles são relevantes em todas as áreas, as três principais características que descrevem o fenômeno denominado como “Cisne Negro” são a raridade, o impacto extremo e a previsibilidade retrospectiva. Pensando nesse conceito para o caso de estudo deste trabalho, poderia ser por exemplo um evento *outlier* como uma catástrofe natural nos *data centers*, ou uma queda de energia que a infraestrutura não tenha chance de suportar, ou um corrompimento de uma parte significativa dos dados. Todos esses casos podem ser responsáveis por causar falhas irreparáveis ou perda de dados em uma grande escala e também, em retrospecto pode ser possível chegar em uma explicação mas não em um modelo ou previsão para acontecimentos futuros desse tipo.

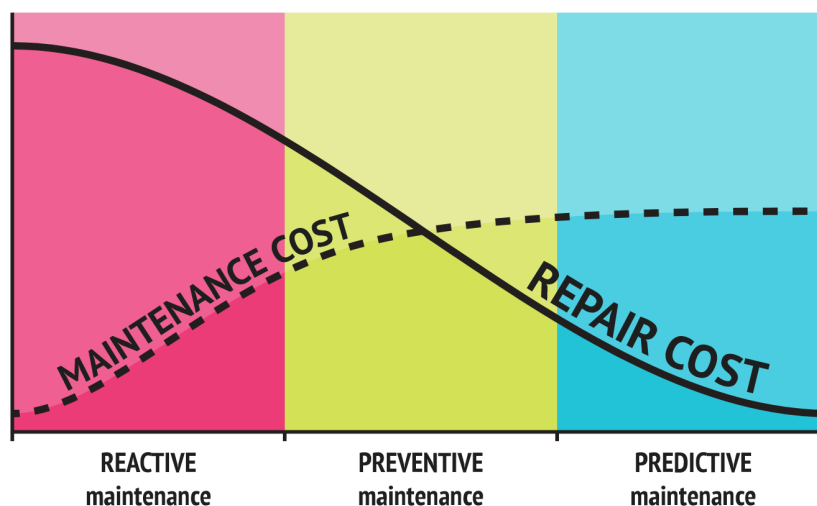
O caso que acabou de ser descrito é um exemplo que, quase por definição, não pode ser previsto, ou é extremamente difícil, pois envolve aleatoriedades reais e sistemas muito complexos. No entanto, os outros eventos, falhas, mudanças que não se encaixam nesse fenômeno podem de alguma forma serem esperados, preparados e até previstos.

De forma mais prática, mais cedo ou mais tarde, todas as máquinas apresentam falhas, mas com uma ampla gama de consequências. De acordo com a altexsoft [32], o custo médio do tempo de inatividade (downtime) não planejado nos setores de energia, manufatura, transporte e outros é de 250.000 dólares (US\$) por hora ou 2 milhões de dólares (US\$) por dia útil.

Dessa forma, é de grande interesse ter medidas para evitar qualquer tipo de *downtime* no serviço. Neste trabalho, o foco é no estudo do uso de Modelos de AM, mais especificamente de árvores de decisão, para prever falhas em discos rígidos (HDs) com os dados provenientes de suas métricas SMART e RUL. Com isso, a área de estudo de Manutenção Preditiva é extremamente relevante por estudar as diferentes formas de lidar com a manutenção de produtos e com as suas falhas. Além disso, também fornece as principais estratégias possíveis para aplicar em um sistema e ferramentas úteis como modelos capazes de calcular o Tempo Restante de Vida de um disco rígido, por exemplo.

Empresas precisam ter uma política de manutenção eficiente de acordo com o produto e equipamentos manuseados para funcionamento normal. Existem três tipos principais de políticas de manutenção que podem ser adotadas: Corretiva, Preventiva e Preditiva. Cada opção tem suas vantagens e desvantagens.

**Figura 8:** Alocação de custos em diferentes cenários de manutenção.



Fonte: [Tibbo Systems](#), 2020

- **Prevenção Reativa [32]**

Trata-se de corrigir o problema quando ele realmente acontece. Muitas vezes aplicados em produtos não reparáveis (c/ MTTF) ou que têm sua substituição por baixo custo e de forma fácil. Também conhecida como manutenção corretiva, *run-to-failure* ou de quebra, significa que as ações são tomadas quando o equipamento já está quebrado. Essa abordagem economiza tempo e dinheiro em planejamento e serviços de suporte. Ela pode ocorrer no caso de ativos redundantes, fáceis de consertar e não críticos para o sistema. Por exemplo, as lâmpadas são substituídas somente depois de queimar.

Embora esse tipo de manutenção não exija custos iniciais, ela acaba sendo muito cara no longo prazo, considerando os salários das horas extras, a redução da vida útil dos ativos, os danos à reputação da confiabilidade do sistema e os riscos à segurança. De

acordo com estimativas do Marshall Institute [33], a abordagem reativa custa às empresas até cinco vezes mais do que os tipos proativos de manutenção. No caso dos HDs, essa abordagem tem tempo significativo para restabelecimento do estado ideal do sistema. Enquanto a falha não for tratada e o disco substituído, a disponibilidade e a durabilidade ficam comprometidas.

- **Prevenção Preventiva [32]**

Nessa abordagem, os consertos estão em uma agenda definida. É mais aplicado em caso de produtos que são em grande parte reparáveis (com MTBF). A manutenção preventiva indica inspeções regulares nos equipamentos para mitigar a degradação e reduzir a probabilidade de falhas. Atividades planejadas, como lubrificação ou troca de filtros, podem prolongar a vida útil dos ativos e aumentar sua eficiência. Tudo isso se traduz em economia de dinheiro. Estudos mostram que a economia média da manutenção planejada chega a 12% a 18% em relação à manutenção reativa [32].

Entretanto, as medidas preventivas não podem excluir totalmente a possibilidade de falhas catastróficas. Além disso, essa prática envolve maior planejamento e recursos humanos adicionais. Muitas vezes, a frequência das verificações é maior ou menor de acordo com a meta de confiabilidade a ser atingida. No caso dos HDs, é uma abordagem pouco usada pois o monitoramento da saúde dos discos já faz o papel de checagem do equipamento sem precisar que ele pare de funcionar. Discos também não são facilmente reparáveis no *data center* e normalmente valem ser trocados de forma rápida e ter os dados remapeados para o novo dispositivo.

- **Prevenção Preditiva [32]**

Essa alternativa tem como foco não consertar o que não está quebrado e para isso utiliza muito da informação de vida útil restante do equipamento, descrita pelo atributo RUL. Além de conhecimentos relacionados ao monitoramento frequente de métricas relevantes ao aparelho em questão, métricas SMART no caso dos HDDs e desenvolvimento de modelos de Aprendizado de Máquina com esses dados podem ser consideradas para prever falhas iminentes.

Semelhante à manutenção preventiva (B), a PdM é uma abordagem proativa para a manutenção de aparelhos. A diferença é que uma empresa programa as atividades, como troca de certos discos em racks, assim que um modelo indicar que o disco específico tem traços de falha iminente ou muitos padrões de falha (alta AFR) em um modelo, com base no monitoramento constante das condições, com o uso de técnicas que serão melhores descritas a seguir. Esses reparos ou substituições feitos antes de uma falha ser

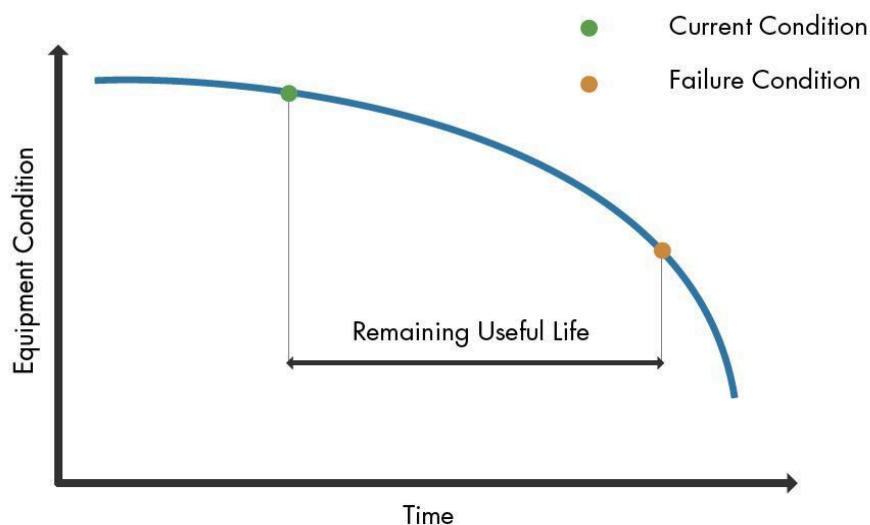
concretizada fazem com que falhas mais caras sejam potencialmente evitadas e o sistema como um todo torne-se mais confiável, disponível e durável.

Os benefícios que a PdM traz para as empresas incluem custos menores nas operações de manutenção, vida útil prolongada do equipamento, tempo de inatividade reduzido, aumento da capacidade de produção e maior segurança. De acordo com um relatório da Deloitte Insights [34], a PdM promete uma redução de 20 a 50% no tempo necessário para planejar a manutenção, um aumento de 10 a 20% no tempo de atividade e na disponibilidade do equipamento e uma redução de 5 a 10% no custo geral de manutenção.

## 4.2 Remaining Useful Life (RUL)

A figura 9 mostra um possível perfil de deterioração de uma máquina ou sistema, com RUL, sendo representado pelo intervalo de tempo entre a condição corrente e a de falha.

**Figura 9** - Exemplo de perfil de deterioração de equipamentos



Fonte: [LinkedIn](#), 2020

### 4.2.1 Entendendo o RUL em HDDs

RUL (*Remaining Useful Life*), ou Vida Útil Remanescente, é uma estimativa de quanto tempo resta antes que um componente, como um disco rígido (HDD), falhe ou se torne inutilizável. Na manutenção preditiva, o cálculo do RUL é fundamental para planejar a substituição ou manutenção do componente antes que ele falhe, evitando perda de dados, tempo de inatividade ou outros problemas críticos.

Para discos rígidos, o RUL é a quantidade de tempo (geralmente medida em horas de operação ou ciclos de uso) que um HDD deve continuar funcionando corretamente antes de falhar. O cálculo do RUL se baseia em dados históricos de falhas de discos similares, indicadores de saúde (como as métricas SMART) e, possivelmente, modelos de degradação específicos.

Enquanto um “Cisne Negro” diz respeito a eventos desconhecidos e não modeláveis, um “Cisne Cinzento” diz respeito a eventos extremos modeláveis. Como por exemplo, a falha de 2 ou 3 discos específicos que contêm todas as réplicas de alguns dados de um usuário em um período de tempo menor que o MTTR. Esse evento tem uma chance extremamente baixa de ocorrer em um *data center* normal, mas é diferente de 0. Sua probabilidade de ocorrer pode ser modelada ao utilizar informações como MTBF, MTTR, Drive Count, Política de Armazenamento que define a estratégia de redundância, Métricas SMART e RUL. Deste modo, o RUL é o componente mais completo e com maior potencial individual de previsão do futuro de um produto, dadas informações históricas e a aplicação correta de um dos modelos que estão explicados a seguir.

#### 4.2.2 Formas de Estimar o RUL

Para estimar o *Remaining Useful Life* [35] de um sistema ou componente é comum utilizar uma das três abordagens: Modelo de Similaridade, Modelo de Sobrevivência ou Modelo de Degradação. A principal diferença entre essas abordagens está nos dados disponíveis para treinar um modelo e chegar em um valor adequado para o RUL.

No Modelo de Sobrevivência, os dados são os mais limitados e apenas retratam o momento de falha do componente, sem muitas informações sobre momentos anteriores quando o disco estivesse saudável, por exemplo.

**Figura 10:** Exemplo informação de falha no Modelo de Sobrevivência



Fonte: [Youtube](#), 2019 [35]

Enquanto no Modelo de Degradação existe um conhecimento de um intervalo do funcionamento do aparelho até antes do momento da falha, tendo apenas uma ideia de um limiar (*threshold*) de segurança que não deve ser passado.

**Figura 11:** Exemplo informação de falha no Modelo de Degradação



Fonte: [Youtube](#), 2019 [35]

No Modelo de Similaridade os dados abrangem desde um estado saudável até a falha, dessa forma os discos da Backblaze que são o objeto de estudo principal deste trabalho estão em sua grande maioria nesta categoria.

**Figura 12:** Exemplo informação de falha no Modelo de Similaridade



Fonte: [Youtube](#), 2019 [35]

### 4.3. Modelos de AM para Predição

Para prever falhas em HDDs, diversos modelos de aprendizado de máquina podem ser aplicados para identificar padrões nos dados que possam indicar uma falha iminente. Os dados utilizados por esses modelos variam consideravelmente, mas em grande maioria são classificados com dados censurados, sejam eles censurados à direita (com limite inferior), à esquerda (com limite superior) e de intervalo (última observação antes da falha e primeira após).

No contexto da engenharia de confiabilidade, normalmente nos os eventos são referidos como “falhas” (*failures*). Em outros setores, uma série de terminologias pode ser usada para descrever eventos. No caso da Biologia Computacional e na aplicação de AM a

casos médicos, o termo “mortes” é usado ao estudar seres vivos.

Levando em conta essa intersecção de domínios que recebem a aplicação de modelos de AM similares para dados com comportamentos e distribuições também similares, é bastante relevante analisar algumas das principais estratégias utilizadas nas áreas de Biologia Computacional para extrair aprendizados e comparar com abordagens da área de estudo deste trabalho.

O artigo “Ten quick tips for machine learning in computational biology” [36] é o foco de estudo desta sessão e alguns de seus ensinamentos foram aplicados ao longo do trabalho na parte de replicar o experimento. Essa questão pode ser explicada pois, nele há diversas técnicas relevantes apresentadas para serem aplicadas aos dados dos discos rígidos. Além do mais, de forma similar a um dataset da biologia, os discos se comportam como uma população e as métricas indicam a saúde dos discos. Existindo casos de uma população com uma doença e, apesar de que grande parte da população não está doente, é bem relevante que exista uma forma consistente de identificar os indivíduos doentes, sendo possível diagnosticá-los com alta taxa de acurácia, precisão e MCC. Para isso, é necessário que os dados, que por natureza são desbalanceados nesse tipo de problema, sejam corretamente balanceados, e então tratados para servirem de treinamento para o modelo adequado de aprendizado de máquina. Após a aplicação do modelo, alguns pontos devem ser avaliados e checados no que se diz respeito ao modelo não estar gerando muitos falsos negativos, ou seja, preferindo a classe majoritária que não tem a doença em prol de aumentar a acurácia, por exemplo.

Alguns dos modelos que podem ser considerados relevantes no funcionamento de discos são apresentados a seguir.

#### 4.3.1 Decision Trees (Árvores de Decisão)

- Árvores de decisão são algoritmos que tomam decisões em formato de árvore, dividindo repetidamente o conjunto de dados em subconjuntos baseados em condições de atributos dos dados. Cada nó da árvore representa uma decisão (ou teste) sobre um atributo, e cada ramo representa o resultado do teste. As folhas representam as previsões finais.
- No caso dos discos sob análise, as árvores de decisão podem usar os atributos SMART, RUL, temperatura, e histórico de erros para criar regras de decisão que separam os HDDs que falham daqueles que não falham, como tratado no trabalho “Interpretable predictive maintenance for hard drives” [1]

### 4.3.2 Random Forest

- *Random Forest* é uma técnica que utiliza uma combinação de múltiplas árvores de decisão, onde cada árvore é treinada com uma amostra aleatória do conjunto de dados. O resultado final é determinado pela média ou espécie de votação da previsão de todas as árvores. A ideia é que o conjunto de árvores pode corrigir os erros de uma árvore isolada, tornando o modelo mais robusto e menos propenso a overfitting (ajuste excessivo). Como resultado, comumente permite uma melhor generalização e maior precisão, especialmente em situações com muitos atributos correlacionados.

### 4.3.3 Gradient Boosting

- *Gradient Boosting* é um algoritmo de *boosting* utilizado para problemas de regressão e classificação, que produz um modelo de previsão na forma de uma combinação (*ensemble*) de modelos de previsão considerados fracos, como árvores de decisão, em que cada árvore é treinada para corrigir os erros da árvore anterior. Esta técnica minimiza uma função de perda através de gradientes, ajustando os pesos dos dados de treinamento de acordo com a dificuldade em prever corretamente. Como resultado, é altamente eficiente, pois o modelo ocorre em etapas, como outros métodos de *boosting*, e os generaliza, permitindo utilizar otimizações computacionais e conseguindo lidar com dados esparsos. Em geral, é mais eficaz quando há muitos atributos e interações não lineares.

### 4.3.4 Deep Learning (Redes Neurais)

- Aprendizado profundo é um termo amplo, que abrange diversos algoritmos que utilizam redes neurais artificiais juntamente com camadas ocultas, em que cada camada apresenta uma representação. Dessa forma, é possível ter representações de funções complexas de uma maneira mais compacta.
- Redes neurais artificiais são inspiradas na forma de funcionamento do cérebro humano e são compostas de camadas de neurônios conectados entre si. Cada neurônio recebe uma entrada e têm pesos e vieses (*bias*) que serão aplicados, passando por uma função de ativação e gerando uma saída. Em redes profundas (*deep learning*), há múltiplas camadas ocultas que permitem à rede aprender representações complexas e extrair características avançadas dos dados. Quanto maior a quantidade de dados disponíveis, mais eficazes tendem a ser e também são adequadas para dados que possuam relações não lineares e complexas.

### 4.3.5 SVM (Support Vector Machines)

- *Support Vector Machines* (SVM) são algoritmos de classificação que buscam encontrar um hiperplano ótimo que separa diferentes classes nos dados. SVM tenta maximizar a margem entre as diferentes classes (por exemplo, HDDs com falha e saudáveis), utilizando vetores de suporte que representam pontos mais próximos ao hiperplano de separação. SVM pode ser aplicado quando os dados são de alta dimensão e há uma clara separação entre classes. No entanto, se as falhas de HDDs não forem linearmente separáveis, a técnica SVM pode tentar usar um kernel não linear para projetar os dados em um espaço dimensional superior. Mesmo assim, pode não ser muito ideal para muitas tarefas e tipos de dados.

As três primeiras técnicas citadas, DT, Random Forest e XGBoost (um tipo de Gradient Boosting), são as que apresentam maiores valores de acurácia na predição, tendo sido observados valores acima de 0.8 para esses três [30] em estudos relacionados. Considerando isso, esses foram os três modelos implementados neste trabalho para serem comparados entre si, com o foco ainda no modelo de árvore de decisão (DT).

Quando se trata de prever resultados contínuos, como a vida útil restante de um disco, uma análise de regressão padrão pode não parecer totalmente adequada para resolver o problema. Isso ocorre porque é comum que a maioria dos registros nos dados não tenha sofrido uma falha e, portanto, não tenha nenhuma medida ou medidas limitadas de sua vida útil restante [36]. No entanto, ainda é possível usar esses dados “incompletos” para aprendizado, explorando o fato de que ao observar que a grande maioria das máquinas não apresentam falhas por algum tempo é possível determinar limites mais baixos para a vida útil restante, como uma espécie de *threshold*. Por exemplo, se uma máquina sem falhas for observada por 90 dias, é de conhecimento que, a partir do primeiro dia, ela tem uma vida útil restante de pelo menos 90 dias.

A análise de sobrevivência é uma classe especializada de modelos projetados para lidar com os chamados dados censurados e é fortemente resolvida com o uso de modelos de aprendizado de máquina como os vistos anteriormente. Uma ferramenta clássica de análise de sobrevivência é o modelo de riscos proporcionais de Cox, que modela o impacto das características na vida útil usando uma abordagem análoga à regressão linear (Cox, 1972). Outra abordagem tradicional é o estimador de Kaplan-Meier, que é um modelo não paramétrico de sobrevivência ao longo do tempo (Kaplan & Meier, 1958). Os modelos de árvore de decisão também podem ser aplicados a problemas de sobrevivência para aumentar o poder e a interpretabilidade

## Capítulo 5. Desenvolvimento do Trabalho

Considerando os aspectos tratados ao longo deste estudo, o uso de técnicas de tratamento de dados e de IA para criar modelos capazes de detectar as falhas em disco apresentadas, este capítulo descreve como os dados históricos coletados na plataforma Backblaze sobre os discos ST4000DM000 foram usados, desde suas obtenções via download via Internet até o momento de geração e avaliação do modelo de Árvore de Decisão (DT). Deste modo, replica-se, com algumas variações, os experimentos descritos no artigo de referência deste trabalho [1].

Primeiramente, foi preciso selecionar um subconjunto dos dados de entrada que iriam compor o banco de dados. Embora o esquema de banco de dados [7] possa mudar a cada trimestre, informações básicas como *date*, *serial\_number*, *model*, *capacity\_bytes*, e *failure* não mudam. As mudanças se restringem ao número de atributos SMART reportados para todos os discos de um dado trimestre. O número de pares de atributos SMART nunca ultrapassa 255. Nesse estudo, foi necessário tomar em conta essas e outras informações e a possibilidade de haver diferenças nas quantidades de atributos SMART. No caso, a partir desses dados na plataforma Backblaze, os períodos de 2016, 2017 e 2018 foram escolhidos, dentre outros motivos, por apresentarem a mesma estrutura e quantidade de atributos SMART monitorados.

Esse período de 3 anos, com início em 2016 e fim em 2018, também foi o foco do estudo pois o modelo de disco rígido mais utilizado, e também o que apresentava maior número absoluto de falhas de disco, foi um da Seagate de 4TB. A Seagate, como foi pontuado anteriormente pela equipe da Backblaze, também produz HDDs que, em geral, apresentam boa sinalização de falhas iminentes através das estatísticas SMART geradas pelos discos. Por último, esse tamanho de disco não é o mais usado em *data centers* de grande porte, como é o caso da Backblaze, tendo sido ultrapassado pelo modelo ST12000NM0007, Seagate de 12TB, no terceiro trimestre de 2018 [38].

Apesar disso, o modelo ST4000DM000, ou outros de tamanhos similares, ainda é de grande utilidade para *data centers* de menor porte. Até na Backblaze, em que se manteve como um dos modelos mais utilizados por anos, pois ele já estava em operação e com bom custo-benefício, apresentando grande tempo de vida e menor consumo de energia em comparação a modelos maiores. Esse segundo ponto é um fator muito relevante na montagem dos racks com drives heterogêneos e que precisam manter um consumo de energia balanceado.

Neste trabalho, buscou-se inicialmente realizar um estudo dos discos utilizados em *data centers* para a tarefa de armazenamento na nuvem. Como a Backblaze tem a maior quantidade de dados desse tipo publicamente disponíveis para uso, o estudo foi feito em cima de um de seus modelos de discos mais usados até o ano de 2018, o ST4000DM000. Além de tratar-se do modelo com maior ocorrência (*drive count*) até 2018, também é o modelo com maior número absoluto de falhas e taxa de falha na faixa de 2% ou 3%.

Considerando os arquivos disponibilizados no site da Backblaze de 2016 para frente, os dados encontram-se em um arquivo em formato .csv para cada dia do ano, e apresentam tamanhos variando de 1.42 GB até 10.16 GB, após descompressão. Cada arquivo engloba os dados obtidos de 90 métricas SMART, sendo metade normalizada e metade *raw* (bruto), de cada disco instalado e funcionando em um dia. No entanto, algumas das métricas apresentam valores nulos para discos específicos e os atributos mais relevantes para predição de falhas podem variar de modelo para modelo.

Os atributos mais relevantes, no quesito de correlacionar com falhas, de acordo com estudos feitos, tanto aqui no trabalho quanto em outros trabalhos [1] [21] [22], foram:

- SMART 5: Reallocated\_Sector\_Count
- SMART 187: Reported\_Uncorrectable\_Errors
- SMART 188: Command\_Timeout
- SMART 197: Current\_Pending\_Sector\_Count
- SMART 198: Offline\_Uncorrectable\_Sector\_Count
- SMART 1: Read Error Rate
- SMART 3: Spin-Up Time
- SMART 7: Seek Error Rate
- SMART 10: Spin Retry Count
- SMART 184: End-to-End error / IOEDC
- SMART 190: Temperature Difference
- SMART 196: Reallocation Event Count

Totalizando 12 pares de atributos SMART diferentes e que são considerados mais importantes para predizer falhas em HDDs..

A partir deste momento, todas as figuras e todos os trechos de códigos mostrados e citados têm como fonte os próprios trabalhos do autor deste texto. O Notebook Python desenvolvido está disponibilizado na referência e os principais trechos de códigos não apresentados aqui no trabalho estão presentes no apêndice, que também pode ser encontrado na referência.

Primeiramente, os 12 arquivos trimestrais compactados, 4 por ano por 3 anos, contendo estatísticas de discos presentes no site da Backblaze [7] foram baixados. Em seguida, foram extraídos para um arquivo que contém uma separação por ano e foi feita uma transformação dos diversos arquivos .csv para um arquivo .db por trimestre, já ocorrendo a seleção de apenas as colunas de interesse. Uma vez que cada trimestre tem a sua própria tabela dentro de um mesmo arquivo .db (20??disk.db), o programa DB Browser For SQLITE (DB4S) [27] foi utilizado para abrir as tabelas e checar qualquer possível irregularidade. A principal irregularidade encontrada foi o fato do atributo 196 apresentar valores nulos para esse disco da Seagate de 4TB e por isso foi retirado na etapa de transformar os múltiplos arquivos .csv do trimestre em um só arquivo .db. Deste modo, a base de dados resultante passou a possuir 11 pares de atributos SMART diferentes.

Com esse enfoque nos dados de um modelo de disco e nas métricas mais relevantes, foi possível diminuir significativamente o tamanho dos arquivos .csv e criar um arquivo .db com uma tabela de dados de cada trimestre, totalizando 4 tabelas por arquivo e 3 anos (2016, 2017 e 2018), mas que ainda tratavam de todos os modelos de disco na infraestrutura de armazenamento da Backblaze. Após isso, a linguagem SQL foi utilizada para criar tabelas que tivessem apenas o modelo de HD de interesse ST4000DM000 a partir do arquivo .db, reduzindo ainda mais o tamanho do arquivo e limitando-o apenas aos dados de interesse utilizados para o treinamento do modelo de aprendizado de máquina para prever falhas no disco.

Na figura 13, é possível visualizar as tabelas trimestrais do ano de 2016 geradas com SQL sendo combinadas por meio de uma view que realiza “UNION ALL” dos atributos de interesse em uma grande tabela, que pode ser vista na Figura 14 logo antes de ser exportada para o formato .csv. Deste modo, ao fim da aplicação desses passos nos anos de 2016, 2017 e 2018, como pode ser visto na Figura 15, que mostra que foi gerado um grande arquivo .csv para cada ano de operação em análise. Cada um dos arquivos .csv resultantes, de mais de 1 GB cada, foi carregado no ambiente de desenvolvimento para ser tratado com código python. Os dados servem como *dataframe* e são limpos para treinamento e posterior aplicação do modelo de aprendizado de máquina selecionado.

**Figura 13:** Trecho do código SQL para gerar tabela de interesse

```

1 CREATE VIEW diskModelSeagate4_2016 AS
2 SELECT date, serial_number, failure, smart_3_normalized, smart_3_raw, smart_5_normalized, smart_5_raw, smart_7_normali
3 FROM "main"."2016q1ST_table"
4 WHERE model LIKE 'ST4000DM000'
5 UNION ALL
6 SELECT date, serial_number, failure, smart_3_normalized, smart_3_raw, smart_5_normalized, smart_5_raw, smart_7_normali
7 FROM "main"."2016q2ST_table"
8 WHERE model LIKE 'ST4000DM000'
9 UNION ALL
10 SELECT date, serial_number, failure, smart_3_normalized, smart_3_raw, smart_5_normalized, smart_5_raw, smart_7_normali
11 FROM "main"."2016q3ST_table"

```

**Figura 14:** Tabela dos atributos relevantes do disco ST4000DM000 de 2016

	date	serial_number	failure	smart_3_normalized	smart_3_raw	smart_5_normalized	smart_5_raw	smart_7_normalized	smart_7_raw	smart_187_normalized	smart
1	2016-01-01	Z300KHNO	1	91.0	0.0	100.0	0.0	85.0	401386362.0	100.0	
2	2016-01-01	Z3015V4D	1	92.0	0.0	100.0	0.0	77.0	51339476.0	100.0	
3	2016-01-01	W3006E5B	1	91.0	0.0	100.0	0.0	87.0	474660411.0	100.0	
4	2016-01-02	S301146W	1	97.0	0.0	100.0	0.0	84.0	246239674.0	1.0	
5	2016-01-02	Z3025LCN	1	97.0	0.0	100.0	0.0	86.0	444492981.0	99.0	
6	2016-01-02	Z300GPY9	1	93.0	0.0	100.0	0.0	87.0	649893268.0	100.0	
7	2016-01-02	Z302A14M	1	92.0	0.0	100.0	8.0	86.0	440867202.0	100.0	
8	2016-01-02	Z300WF3E	1	91.0	0.0	100.0	0.0	72.0	20949675.0	99.0	
9	2016-01-03	Z30054Q5	1	92.0	0.0	100.0	0.0	84.0	302198100.0	100.0	
10	2016-01-03	Z3025K64	1	94.0	0.0	100.0	0.0	87.0	606633326.0	100.0	
11	2016-01-04	Z304JD7V	1	98.0	0.0	100.0	8.0	79.0	98470575.0	96.0	
12	2016-01-04	S3007K7N	1	94.0	0.0	100.0	0.0	100.0	962316.0	100.0	
13	2016-01-06	Z3015CR3	1	91.0	0.0	100.0	0.0	85.0	382687592.0	100.0	
14	2016-01-06	Z300X7EZ	1	92.0	0.0	100.0	28.0	85.0	397892113.0	100.0	
15	2016-01-06	W3008SKD	1	92.0	0.0	100.0	0.0	88.0	719021228.0	100.0	
16	2016-01-06	Z300WGAW	1	92.0	0.0	100.0	0.0	86.0	427713589.0	100.0	
17	2016-01-06	W300C8F6	1	92.0	0.0	100.0	0.0	87.0	563878541.0	97.0	
18	2016-01-06	Z303RK24	1	95.0	0.0	100.0	1.0	78.0	71391245.0	1.0	
19	2016-01-06	Z300GZ1N	1	92.0	0.0	100.0	0.0	87.0	657016083.0	100.0	
20	2016-01-07	Z304HT8H	1	99.0	0.0	100.0	0.0	75.0	40171762.0	100.0	
21	2016-01-07	Z300NK93	1	92.0	0.0	100.0	32.0	87.0	606741971.0	91.0	
22	2016-01-07	Z300ZQYJ	1	91.0	0.0	100.0	408.0	84.0	268805112.0	100.0	
23	2016-01-08	W300D36Z	1	92.0	0.0	100.0	0.0	87.0	563921754.0	94.0	
24	2016-01-08	W300H36V	1	92.0	0.0	100.0	0.0	87.0	491192217.0	100.0	
25	2016-01-08	Z304GR9H	1	96.0	0.0	100.0	0.0	81.0	119368040.0	82.0	
26	2016-01-10	Z3014A4S	1	91.0	0.0	100.0	32.0	84.0	316406912.0	100.0	
27	2016-01-10	W30068RN	1	92.0	0.0	100.0	0.0	86.0	443646616.0	100.0	
28	2016-01-10	Z3041L93	1	95.0	0.0	100.0	0.0	82.0	181825345.0	94.0	

**Figura 15:** Arquivos .csv gerados no diretório acessado pelo Python Notebook desenvolvido

```

/ JupyterFolder / threeYearCsv /

```

Name	File Size
diskModelSeagate4_2016_moreCol.csv	1.7 GB
diskModelSeagate4_2017_moreCol.csv	1.5 GB
diskModelSeagate4_2018_moreCol.csv	1.4 GB

Para realizar as análises desejadas, o arquivo .csv de cada ano é aberto no Notebook Python e então é feita uma análise exploratória dos dados. Primeiro, foram checados os atributos presentes no *dataframe* e eles ainda estavam corretos, apenas apresentavam alguns valores nulos em algumas linhas. Esses valores nulos foram então retirados de cada ano separadamente. As colunas de *date* e *serial\_number* também foram retiradas, pois não são informações relevantes para o treinamento do modelo.

**Figura 16:** EDA, exemplo de dados de 2016

```
[13]: file.shape

[13]: (12393606, 23)

[14]: # Dados bastante desbalanceados
file.failure.value_counts()

[14]: failure
0      12392667
1         939
Name: count, dtype: int64

[15]: # Quantidade de discos (serial_number distintos)
len(file0.serial_number.unique())

[15]: 35678
```

Na Figura 16, já é possível ver informações relevantes relacionadas ao *shape* do *dataframe*, à quantidade de drives únicos (35678), às colunas e à quantidade de entradas no banco de dados em que HDs apresentaram falha (1) ou não (0). Desses discos, 939 apresentaram falha no período de 1 ano em 2016, enquanto para o período dos três anos 2574 HDs apresentaram falhas, dado que pode ser conferido na Figura 17.

Em relação ao número de drives únicos em cada ano (*drive count*), o valor foi de 35678 para 2016, 35180 para 2017 e 32164 para 2018. A partir disso, é possível começar a perceber que os HDDs de 4TB, como esse da Seagate, que foram os mais usados por vários anos, estavam começando a serem substituídos por outros modelos com maior densidade de armazenamento.

Após isso, como pode ser visto na Figura 17, os dados já explorados e parcialmente tratados dos três anos foram juntados em um só *dataframe* com 33.702.561 de linhas e 23 colunas. Ao contabilizar o número de atributos *failure* com valor 1 de todos os 3 anos, o valor de 2574 falhas foi atingido. Ou seja, cerca de 2,35 discos da Seagate de 4TB falharam por dia na infraestrutura da nuvem estudada. Já a proporção de discos que falham para todos os discos na infraestrutura, pegando a média de contagem de discos dos três anos desse modelo, foi de 1/13,34.

**Figura 17:** Concatenação dos dataframes dos três anos e contabilização de falhas

```
# Concatenação será usada para os diferentes anos
file = pd.concat([df2016, df2017, df2018],axis = 0)

file.shape

(33702561, 23)

file.failure.value_counts()

failure
0    33699987
1     2574
```

**Figura 18:** Resultado da subamostragem dos dados da classe 0 (discos sem falha)

```
downsampled_df.shape

(10000, 23)

y_resampled.value_counts()

failure
0    7426
1     2574
```

Antes de realizar a subamostragem (*undersampling*), o número de linhas no dataset com valor 0 na coluna de falha era de “33.699.987”. Esse número é explicado ao pensar que todos os discos (*drive\_count*) são monitorados uma vez ao dia e todos os dias de seu funcionamento eles devolvem o valor 0 na coluna de falha, pois estão funcionando, o que leva a pensar na ideia do Viés do Sobrevivente (*Survivorship Bias*) [39]. No entanto, os valores 1 que aparecem no y ou na coluna *failure* do dataset são resultados de uma falha que ocorreu em um disco em um dia. Após isso, ele é retirado de produção e ele para de informar o valor 1.

Levando em conta que esse conjunto de dados utilizado é extremamente desbalanceado, foi aplicada a técnica de *undersampling* na classe majoritária [36], como pode ser visto na figura 18, e, posteriormente, na etapa de divisão dos dados de treino e teste, foi aplicado o *oversampling* na classe minoritária do treinamento do modelo por meio do SMOTE [28]. Em seguida, a Figura 19 mostra a aplicação da normalização dos dados por meio do *StandardScaler* da biblioteca sklearn.

**Figura 19:** Normalização dos dados usando *scaler*

```
Normalização

from sklearn.preprocessing import StandardScaler as ss
scaler = ss()

X = downsampled_df.drop(['failure'],axis = 1)

X = scaler.fit_transform(X)

y = downsampled_df.iloc[:,-1]

y.sample(3)

171525      1
5365910     1
5572945     0
```

Para manter a análise interpretável e intuitiva, foram utilizadas apenas as métricas SMART brutas e normalizadas, capturadas nos dados, e não foi criado nenhum recurso mais complexo. Além disso, tinha-se o objetivo dos modelos preditivos de se concentrar na caracterização da saúde do disco e não em algum atalho que pudesse dar bons resultados.

Dentre os vários atributos SMART retirados na etapa de tratamento de dados, algumas métricas que representam contagens cumulativas ao longo da vida útil do disco e, portanto, são altamente correlacionadas com o tempo (4, 9, 12, 192, 193, 240, 241 e 242), foram também removidas. Isso é relevante notar pois permitiu remover do modelo o resultado óbvio de que os discos mais antigos têm maior probabilidade de falhar, pois é algo que foi tratado no estudo da confiabilidade (2.2) e que é descrito pelas curvas de análise de sobrevivência aplicadas aos HDs, como na Curva da Banheira. Por esse motivo, tornou-se possível descobrir informações preditivas mais interessantes com base em métricas mais relacionadas ao status geral de saúde dos discos, independentemente da idade.

As partes de explicação do modelo e do dataset foram tratadas em capítulos anteriores e a parte do trabalho que envolve treinar, avaliar e modificar o modelo de aprendizado de máquina é abordada no próximo capítulo, que também descreve o processo que levou à escolha de certos modelos, à divisão de dados e à escolha dos parâmetros.

## Capítulo 6. Resultados e Discussão

Embora alguns passos importantes de tratamento de dados e de preparação do *dataframe* já tenham sido apresentados e explicados neste trabalho, alguns pontos valem ser ressaltados.

Todas as imagens e resultados são por padrão referentes ao modelo de disco ST4000DM000, Seagate de 4TB, a não ser se for explicitado de forma contrária. Os discos com valor 0 na variável *y* são aqueles que não apresentaram falha no dia, enquanto os discos com valor 1 em *y* apresentaram um evento de falha em um dia. Em outras palavras, os discos apresentam entradas (linhas) com valor 0 na coluna de *failure* para cada um dos dias que estiveram funcionais e apenas em um único dia o disco pode apresentar o valor 1 que implica em falha. Esse fenômeno é um grande indicativo do *Survivorship Bias* [39], que foi notado em um momento anterior do trabalho, fenômeno esse que deve ser levado em conta para tratar de estudos nas áreas de análise de sobrevivência e manutenção preditiva. Por esse motivo, foi observado também que o desbalanceamento das classes é significativo a ponto de ter sido necessário fazer um *undersampling* bem grande, reduzindo o tamanho dos dados a serem recebidos pelo modelo para apenas 10.000 amostras.

**Figura 20:** Código que divide os dados entre treino e teste

```

from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y, test_size = 0.35,

y_test.value_counts()

failure
0    2559
1     941
Name: count, dtype: int64

y_train.value_counts()

failure
0    4867
1    1633
Name: count, dtype: int64

# Aplicado apenas para dados de Treino
smote = SMOTE(random_state = 26)
X_train_re,y_train_re = smote.fit_resample(X_train,y_train)

y_train_re.value_counts()

failure
0    4867
1    4867

```

Uma vez que os dados do *dataframe* apresentam dimensões mais adequadas para a aplicação do modelo de AM, foi realizada a divisão dos dados entre treino e teste, operação

realizada pela função “*train\_test\_split*” da biblioteca *sklearn*. Algumas das informações mais relevantes são observadas na Figura 20, como por exemplo o *y\_test* tem tamanho de  $\text{test\_size} \times 10000$  e  $y_{\text{train}} = 10000 - y_{\text{test}}$ . Como o *test\_size* é de 0.35, é possível calcular que o número de elementos de *y\_test* é de 3500 elementos ao total, 2559 (0) + 941 (1) no caso.

Uma outra questão interessante é a de que na Figura 20, apesar dos dados terem sofrido subamostragem, eles ainda estavam desbalanceados e apresentando maior quantidade de discos da classe 0 (sem falha). Para resolver essa questão usou-se a técnica de *Synthetic Minority Over-sampling Technique* (SMOTE) [28] nos dados de treinamento, que em tradução já é capaz de explicar que se trata de um técnica de super-amostragem que cria dados sintéticos a partir de representantes da classe minoritária. Essa técnica não é aplicada nos dados de teste, pois o objetivo da etapa da avaliação do modelo, que é apresentada a seguir, é entender o modelo em funcionamento usando dados mais próximos de reais. A função em python é encontrada na última célula de código da Figura 20.

**Figura 21:** Código de Seleção de Atributos

```

Seleção de Atributos

from sklearn.neural_network import MLPRegressor

# Define an autoencoder
autoencoder = MLPRegressor(hidden_layer_sizes=(15,), activation='relu', solver='adam')

# Train the autoencoder for data reconstruction
autoencoder.fit(X_train_re, X_train_re)

# Use the trained autoencoder to reconstruct data
reconstructed_data = autoencoder.predict(X_train_re)

# Calculate reconstruction errors for each feature
reconstruction_errors = np.mean(np.square(X_train_re - reconstructed_data), axis=0)

threshold = np.percentile(reconstruction_errors, 75) # Example: top 20% samples
selected_feature = np.where(reconstruction_errors >= threshold)[0]

selected_feature

array([ 0,  2,  6, 13, 17, 21], dtype=int64)

X_train_selected = X_train_re[:,selected_feature]

X_train_selected.shape

(11900, 6)

```

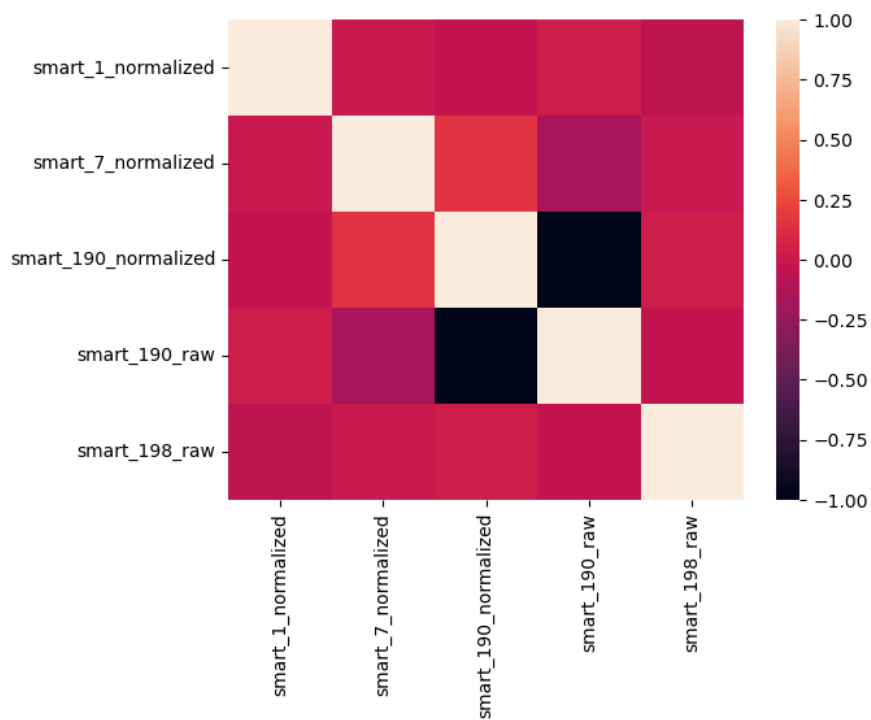
No que se refere aos atributos selecionados, que podem ter o código visualizado na Figura 21, já existe grande progresso feito de antemão no primeiro tratamento de dados. Essa facilidade para chegar em boas *features* só foi possível devido às investigações de artigos relevantes e devido a um estudo amplo sobre as métricas SMART e suas

correlações com falhas de disco. O autoencoder [29] é usado como uma ferramenta não supervisionada para seleção de atributos, identificando as representações de espaço latente que são mais importantes para reconstrução dos dados de entrada. Essas representações de espaço latente podem ser usadas para selecionar um subconjunto de atributos para treinar o modelo de AM. Isso pode melhorar o desempenho do modelo, reduzindo o ruído nos dados e concentrando-se nos recursos mais essenciais. Também é interessante observar a influência da variável *threshold* ao variar o número entre 0 a 100, sendo que em 100, o *threshold* é máximo e nenhuma *feature* seria gerada e, para valores cada vez menores, cada vez mais *features* podem ser selecionadas para treinar o modelo que está por vir. Os resultados provenientes do *autoencoder* para chegar nos melhores atributos são satisfatórios com o valor do *threshold* em 75, significando que os 25% atributos mais relevantes são selecionados.

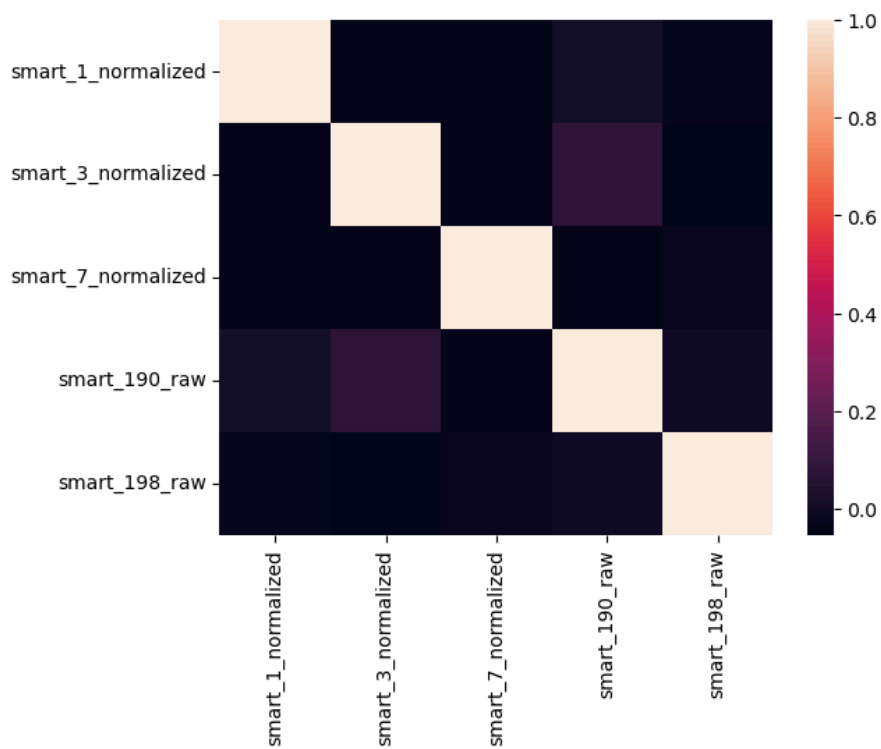
Nesta parte, são mostradas e explicadas algumas Matrizes de Correlação [40] geradas pelo autor com a biblioteca *Seaborn*. Considerando também que foram geradas matrizes para 5 e 6 atributos por vez, Figura 22 e Figura 23, respectivamente, e que buscou-se não ter matrizes com mais de um atributo igual, mesmo que um seja normalizado e outro *raw*.

A matriz de correlação é uma técnica estatística que busca avaliar o nível de correlação de dois atributos utilizados no treinamento. Neste trabalho, a maioria das matrizes de correlação apresentam relações com valores próximos de neutros (0.00), ou seja, não são atributos muito correlacionados entre si, como pode ser visto nas cores mais próximas do vermelho na Figura 22 e nos tons mais frios e próximos do preto na Figura 23. As principais exceções estão nos atributos de mesmo número SMART com variação apenas em sua representação: *normalized* e *raw*. Métricas como as SMART 190 da figura 22 são quase que equivalentes, por apresentarem alta correlação negativa entre si, isso, pois de certa forma são o mesmo dado representados um com valores absolutos e outro com valores normalizados.

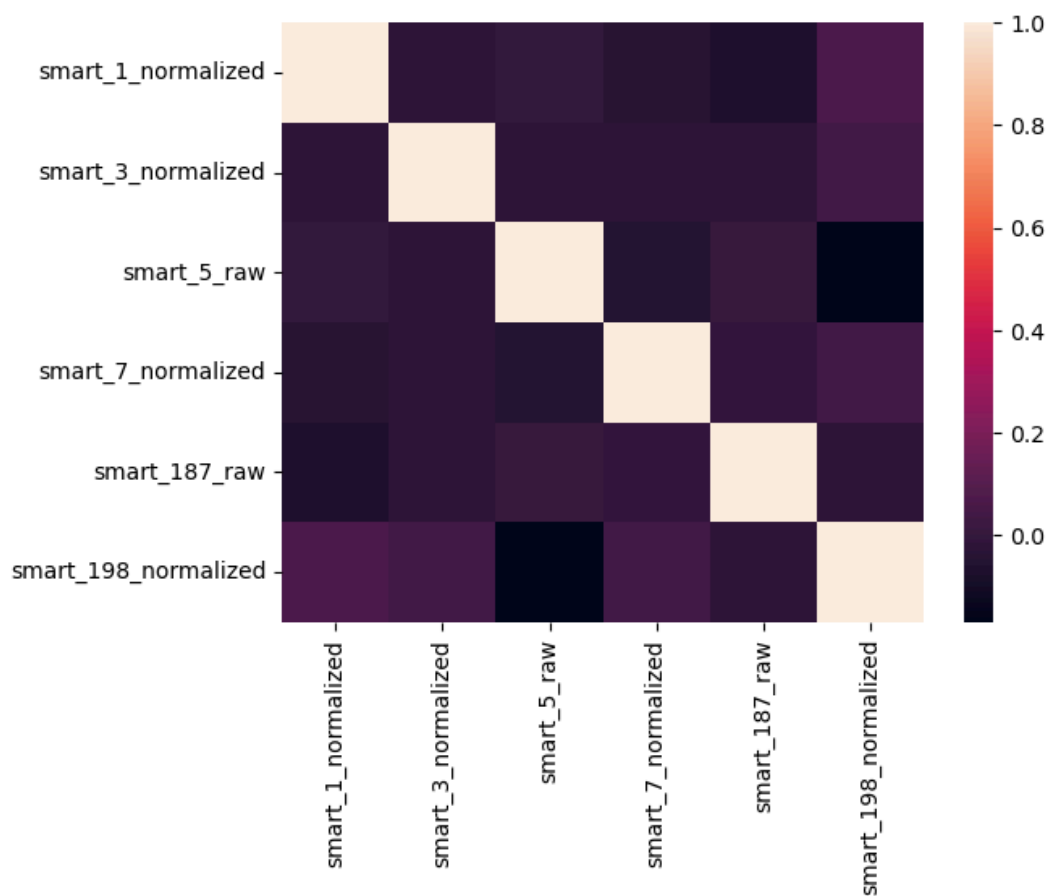
**Figura 22:** Matriz de Correlação de 5 métricas SMART, dados de 2016.



**Figura 23:** Matriz de Correlação de 5 métricas SMART, dados dos 3 anos. Seagate 4TB



**Figura 24:** Matriz de Correlação de 6 métricas SMART, dados de 3 anos.



Na figura 24, pode ser vista uma matriz de correlação com 6 atributos em vez de 5 e nenhuma redundância de métricas. Métricas essas que foram discutidas e corroborando para essa tese como realmente sendo ótimos indicadores de falhas em disco. Todos os valores de cada correlação 2x2 das métricas estão próximos um do outro entre 0.05 e 0.2, em que o valor mínimo é obtido da correlação da métrica 5 raw (*Reallocated Sector Count*) e 198 normalizada (*Offline\_Uncorrectable*). Enquanto o valor máximo obtido de duas métricas nessa matriz é a proveniente dos atributos 1 (*Read Error Rate*) ou 7 (*Seek Error Rate*) com o atributo 198, todos normalizados.

Embora a análise de correlação seja útil para identificar relações entre variáveis, é importante observar que a correlação não implica necessariamente em causalidade. Dois fatores que aparentam variar juntos não significa que um fator cause mudanças no outro. Pode haver uma terceira variável responsável por influenciar ambos. Mas, de forma geral, os valores da matriz da Figura 24 são próximos e quase neutros pois trata-se de métricas que em geral apontam características bem diferentes no disco e que não são diretamente ou fortemente correlacionadas como um todo.

**Figura 25:** Código do Modelo de DT, do sklearn, que está sendo treinado e utilizado

```

from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, precision_score, matthews_corrcoef, confusion_matrix
import pandas as pd
from sklearn import metrics
from sklearn.tree import export_text, plot_tree
import matplotlib.pyplot as plt
dtc3 = DecisionTreeClassifier(max_depth=3)
dtc5 = DecisionTreeClassifier(max_depth=5)

conf = None
def train_classifier(dtc,X_train,y_train,X_test,y_test):
    dtc.fit(X_train,y_train)
    y_pred = dtc.predict(X_test)

    accuracy = accuracy_score(y_test,y_pred)
    print("Accuracy:", accuracy)

    precision = precision_score(y_test,y_pred)
    print("Precision:", precision)

    # Calculate the Matthews Correlation Coefficient
    mcc = matthews_corrcoef(y_test, y_pred)
    print(f'Matthews Correlation Coefficient (MCC): {mcc:.5f}')

    conf = confusion_matrix(y_test, y_pred)

    print(conf)
    # Display the confusion matrix using Seaborn heatmap for better visualization
    plt.figure(figsize=(8,6))
    sns.heatmap(conf, annot=True, fmt='d', cmap='Reds', xticklabels=True, yticklabels=True)
    plt.xlabel('Predicted Labels')
    plt.ylabel('True Labels')
    plt.title('Confusion Matrix')
    plt.show()

# Árvore de Decisão 'DT': dtc
train_classifier(dtc5,X_train_selected,y_train_re.values,X_test_selected,y_test.values)

```

Como dito anteriormente, diversos desafios surgiram enquanto o trabalho estava sendo desenvolvido, principalmente relacionados à área de tratamento de dados e estudo aprofundado para entender os dados e como utilizá-los. Todo o cuidado e trabalho realizado nos dados até o momento definem em grande parte o produto final e os resultados oriundos do modelo em questão. O principal modelo de Aprendizado de Máquina aplicado foi o de Árvore de Decisão ou *Decision Tree* (DT) e ele tem como objetivo resolver o problema de classificação binária de discos saudáveis ou com falha com base em suas métricas SMART. Além desse, os modelos de *Random Forest* e *Gradient Boosting* também foram testados para realizar uma comparação, considerando que esses dois já apresentaram bons resultados em outros artigos.

Na parte da função de *train\_classifier*, na Figura 25, os dados *X\_train* e *y\_train* que sofreram *split* anteriormente são usados como entrada para a função *fit*. Essa função treina a Árvore de Decisão em cima do dataset dos discos e, em seguida, com a função *predict*, pega *X\_test* como entrada e coloca na variável *y\_pred*, que são as amostras com classe prevista como com falha ou não.

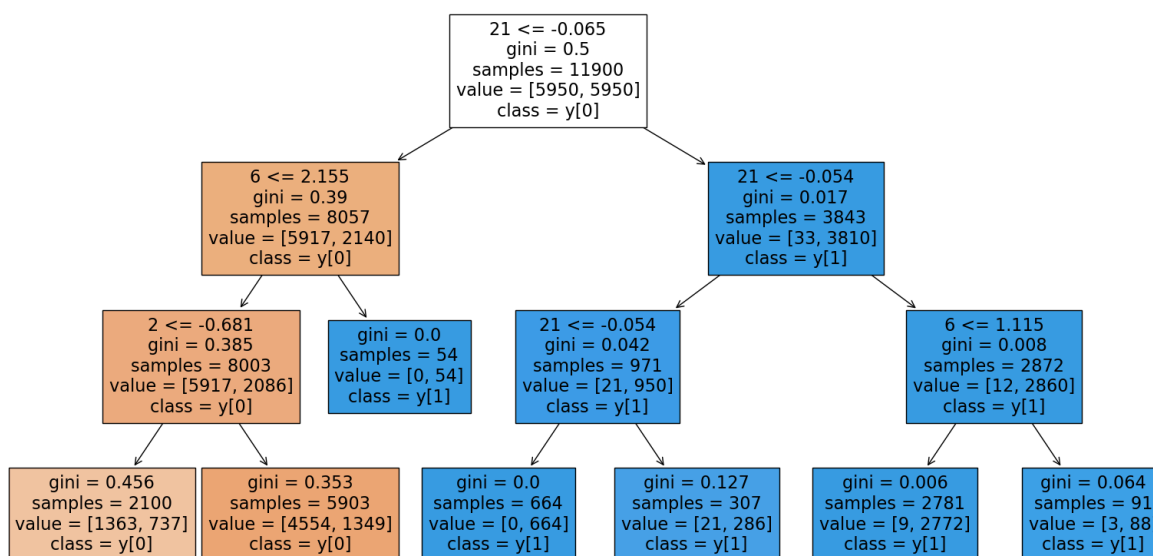
Posteriormente, essa variável *y\_pred* é utilizada em comparação com *y\_test*, que tem os resultados reais dos discos que falharam ou não. Essa comparação é capaz de gerar informações relevantes sobre a efetividade do modelo e de boa parte do trabalho feito até o momento.

A seguir, nas Figuras 26, 27 e 28, são apresentados variações diferentes do mesmo modelo de DTC aplicados aos dados de discos, alguns com diferença nos atributos selecionados ou na profundidade máxima da árvore. Para que exista uma maior clareza e correta interpretação das árvores geradas, foi construída a Tabela 1 para colocar a equivalência dos valores dos atributos utilizados para comparação e para divisão dos dados nos nós da árvore com as métricas SMART correspondentes.

**Tabela 1** - Significado dos valores das DTC (atributos SMART)

0 - Read Error Rate (1 norm)	1 - Read Error Rate (1 raw)
2 - Spin-Up Time (3 norm)	3 - Spin-Up Time (3 raw)
4 - Reallocated_Sector_Count (5 norm)	5 - Reallocated_Sector_Count (5 raw)
6 - Seek Error Rate (7 norm)	7 - Seek Error Rate (7 raw)
8 - Spin Retry Count (10 norm)	9 - Spin Retry Count (10 raw)
10 - End-to-End error (184 norm)	11 - End-to-End error (184 raw)
12 - Reported_Uncorrectable_Errors (187 norm)	13 - Reported_Uncorrectable_Errors (187 raw)
14 - Command_Timeout (188 norm)	15 - Command_Timeout (188 raw)
16 - Temperature Difference (190 norm)	17 - Temperature Difference (190 raw)
18 - Current_Pending_Sector_Count 197 norm	19 - Current_Pending_Sector_Count 197 raw
20 - Offline_Uncorrectable_Sector_Count 198 norm	21 - Offline_Uncorrectable_Sector_Count 198 raw

**Figura 26:** Resultado Treinamento de Árvore de Decisão com 5 atributos e max\_depth=3



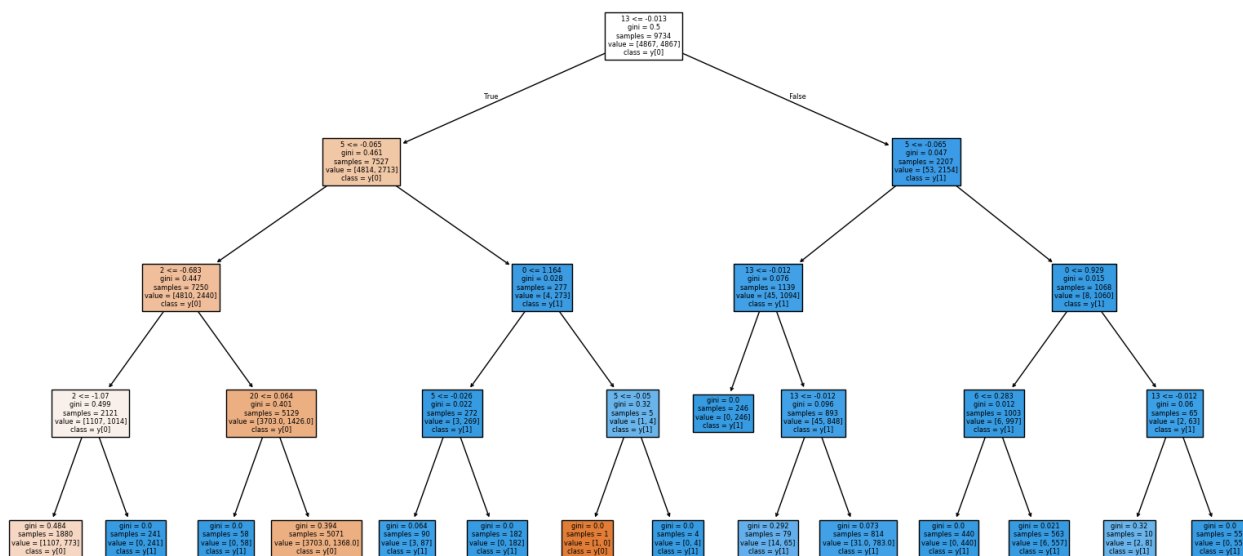
Observando a Figura 26, uma possível interpretação da árvore de decisão gerada começa pelo topo da árvore, em que no primeiro nó é feita uma divisão dos dados, originalmente 11900, levando em conta o atributo 21 do modelo, SMART 198 *Offline Uncorrectable Sector Count*, que indica danos substanciais na superfície do disco ou falhas mecânicas. Quando o valor deste atributo é elevado, há grande indicação de falha nos discos. Assim, ao considerar todos os discos que apresentam o valor normalizado do atributo 21 maior do que -0.065 (lado direito da árvore), isso resultou em 3843 discos sendo classificados como falhos (nós folhas com cor azul).

Considerando então os casos em que o valor do atributo 21 é menor do que -0.065, o próximo nó faz a comparação do atributo 6, SMART 7 *Seek Error Rate*, que contabiliza *seeks* que não foram feitos corretamente na primeira tentativa. Com essa comparação, discos que apresentaram valores maiores que 2.155 para esse atributo também foram classificados como discos com falha, porém foram apenas 54 dos 8057. Todos os outros discos foram classificados como discos sem falha, indicando um ponto fraco específico dessa árvore gerada em que 2086 discos foram classificados sem falha mesmo sendo discos com falha. Isso provavelmente se deve ao fato do parâmetro max\_depth ter sido pequeno ou por conta de não haver um único atributo SMART capaz de separar muito bem os dados no lado esquerdo e na profundidade 2 da árvore, de modo a melhorar isso.

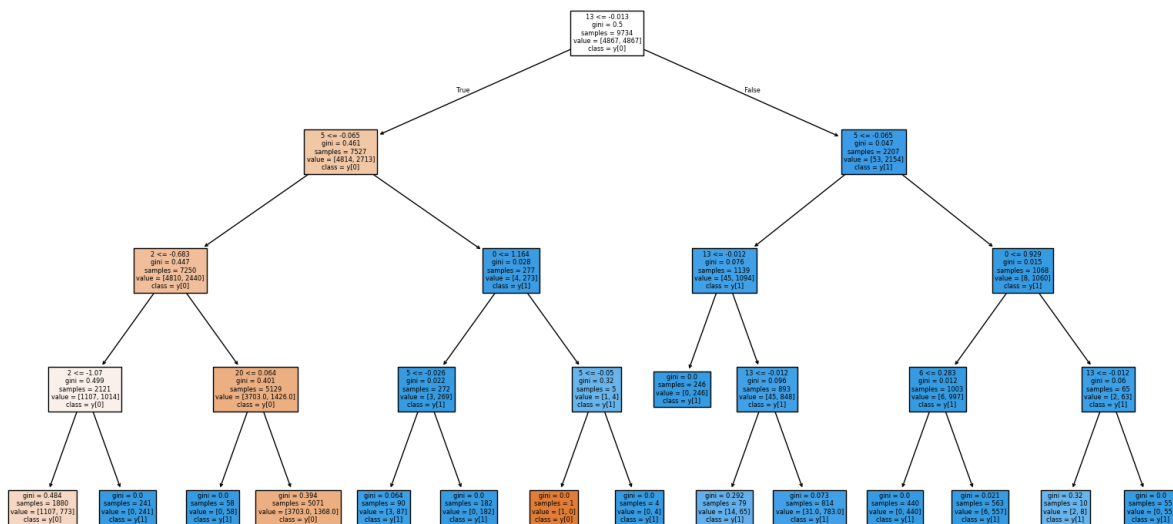
Essa interpretação da árvore de decisão é interessante para entender como a árvore divide os dados de acordo com atributos específicos, porém foi feita em uma árvore pequena, com o intuito de exemplificar o processo de tirar uma interpretação do resultado. Esta não foi, contudo, a árvore que apresentou melhor acurácia, precisão e MCC, que são as principais métricas de avaliação discutidas no trabalho. Um aspecto que pode ser

observado nesta árvore, tratado posteriormente, nas conclusões, é o fato que outras medidas, como o RUL, podem ser combinadas com as métricas SMART para obter melhores resultados.

**Figura 27:** Resultado Treinamento Árvore de Decisão com 6 atributos e max\_depth=4



**Figura 28:** Resultado Treinamento DT com 6 atributos e max\_depth=5



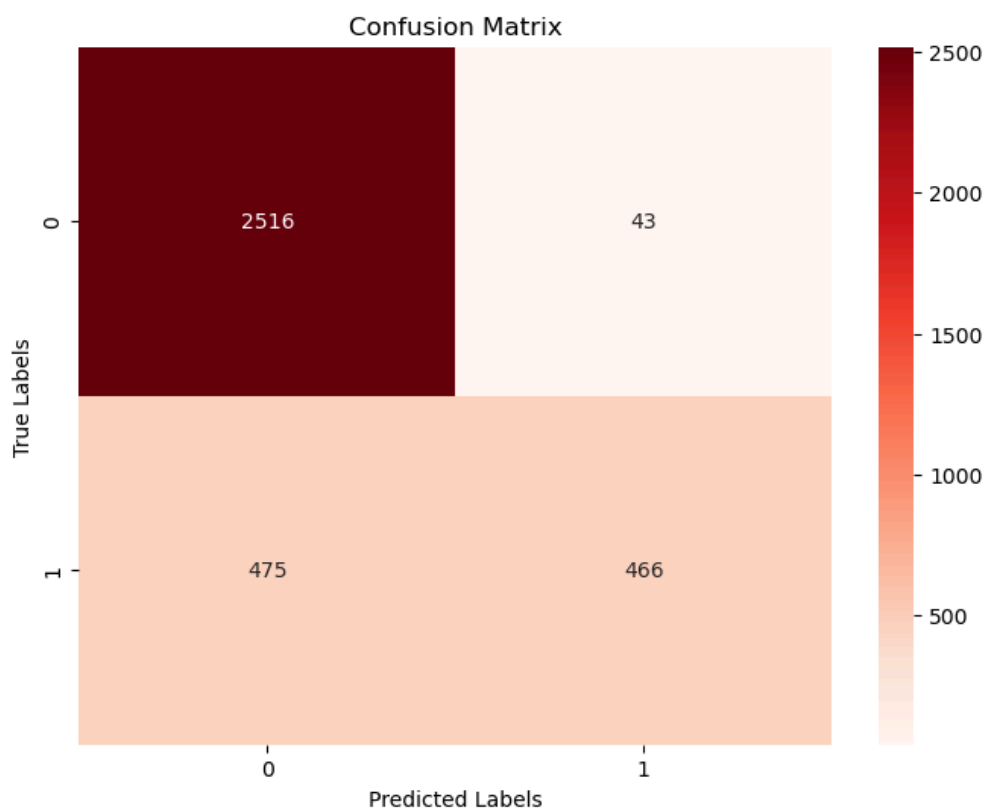
Essas árvores geradas trazem um modelo com capacidade de classificar de forma binária se um disco falhou no dia, dados os seus atributos SMART. Além disso, a DT como classificador tem a grande vantagem de ser uma abordagem de Aprendizado de Máquina

com bons resultados porém sem perder a interpretabilidade do modelo. Isso permite que, qualquer pessoa com um simples entendimento do funcionamento da árvore de decisão e simples consultas para saber qual atributo SMART o nó da árvore está avaliando em cada profundidade consiga entender o que está por trás do modelo e até criar guias para prever de uma forma similar as falhas em discos desse tipo sem o uso de programas.

Após a função *train\_classifier* ter sido chamada, na qual ocorre o treinamento e predição do modelo, além disso, dentro do bloco de código, as principais métricas de avaliação também são calculadas e apresentadas. Essas métricas são obtidas a partir da matriz de confusão, sendo elas a acurácia, precisão e MCC. Por fim, foi feita uma análise de cada uma dessas métricas, com foco na matriz de confusão gerada e, para exemplificar essa análise, será utilizada a matriz de confusão proveniente da Árvore de Decisão gerada na Figura 28.

A matriz de confusão (também chamada de matriz de erro ou tabela de confusão) possibilita visualizar a confusão que um modelo faz ao predizer as classes de um problema. É uma tabela com duas linhas e duas colunas que relata o número de falsos positivos (FP), falsos negativos (FN), verdadeiros positivos (TP) e verdadeiros negativos (TN) [41]. Todos os dados que compõem essa matriz de confusão foram utilizados para avaliar o classificador.

**Figura 29:** Matriz de Confusão de um DTC



Tomando a matriz da Figura 29 como base, cada quadrante da matriz quantifica os dados que encontram-se naquela linha (*True Labels*) e coluna (*Predicted Labels*), sendo que 0 representa um disco sem falha e 1 representa o disco com falha.

**TP - True Positive:** Previsões correta, classe predita é da classe real. (1,1)

TP = 466

**FP - False Positive:** Previsão que é da classe 1 mas não é. (1,0)

FP = 43

**TN - True Negative:** Previsão correta, classe não é a predita. (0,0)

TN = 2516

**FN - False Negative:** Previsão que é da classe 0, mas não é. (0,1)

FN = 475

$$TP + FP + TN + FN = 3500$$

Esse é o valor total de amostras do  $y_{\text{test}}$ , pois houve a definição do  $\text{test\_size}=0.35$  na função de  $\text{train\_test\_split}$ .

- **Acurácia** pode ser calculada com o número de previsões corretas, TP, dividido pelo número total de previsões
  - $(TP + TN) / (TP + FP + TN + FN) = 466$
  - $(2516 + 466) / (2516 + 475 + 466 + 43) = 2982/3500 = 0.852$
- **Precisão** é calculada pelo número de previsões corretas, TP, dividido por todos que foram marcados como sendo da classe em questão, TP e FP
  - $466 / (466 + 43) = 0.916$
- **MCC - Matthews correlation coefficient** é uma medida de avaliação de classificações binárias (duas classes), que varia de -1 a +1.

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}} \quad (10)$$

- Uma pontuação de +1 representa uma previsão perfeita, 0 uma previsão aleatória média e -1 uma previsão inversa.
- Resultado do MCC nesse caso é de 0.602, o que indica uma boa predição.

**Figura 30:** Resultado da função `train_classifier` com a *Decision Tree*

```
Accuracy: 0.852
Precision: 0.9155206286836935
Matthews Correlation Coefficient (MCC): 0.60168
```

A Figura 30 apresenta o output do código que comprova os cálculos realizados com base nas informações da Matriz de Confusão do modelo de Árvore de Decisão.

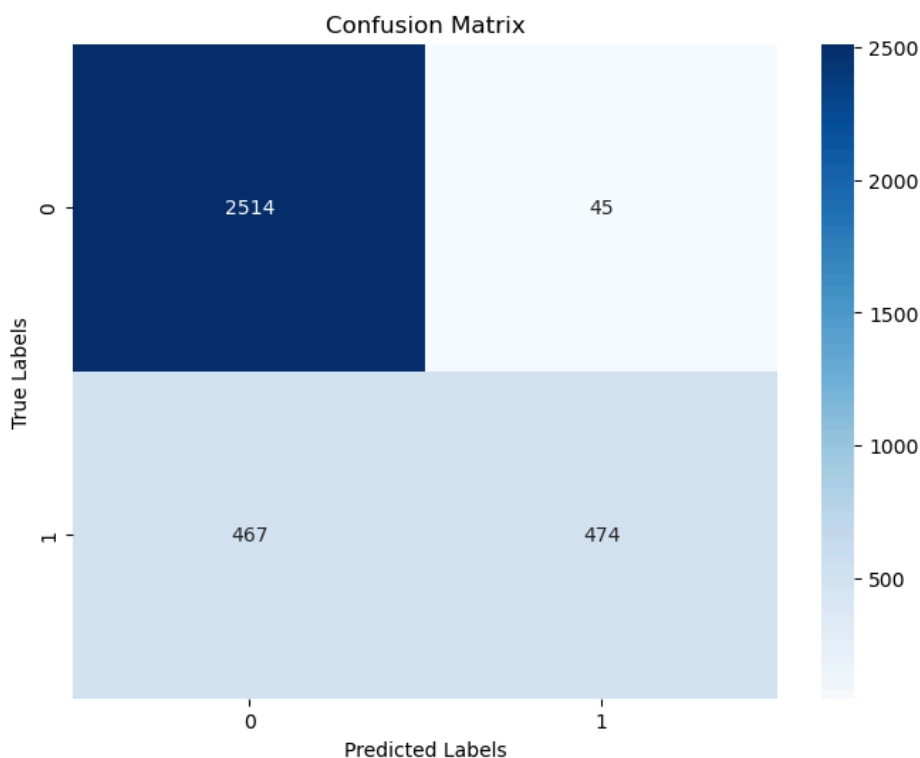
A seguir, nas Figuras 31, 32, 33 e 34, estão os resultados dos modelos de Random Forest e Gradient Boosting, que foram implementados nos mesmos dados para comparar ao modelo foco deste trabalho que são as DTs para a tarefa de predição de falha em discos rígidos.

**Figuras 31-** Resultado da aplicação do Random Forest nos mesmo dados

```
# RandomForestClassifier usage:
train_classifier2(rfc, X_train_selected, y_train)

Accuracy: 0.8537142857142858
Precision: 0.9132947976878613
Matthews Correlation Coefficient (MCC): 0.60649
```

**Figura 32:** Matriz de confusão de um RFC



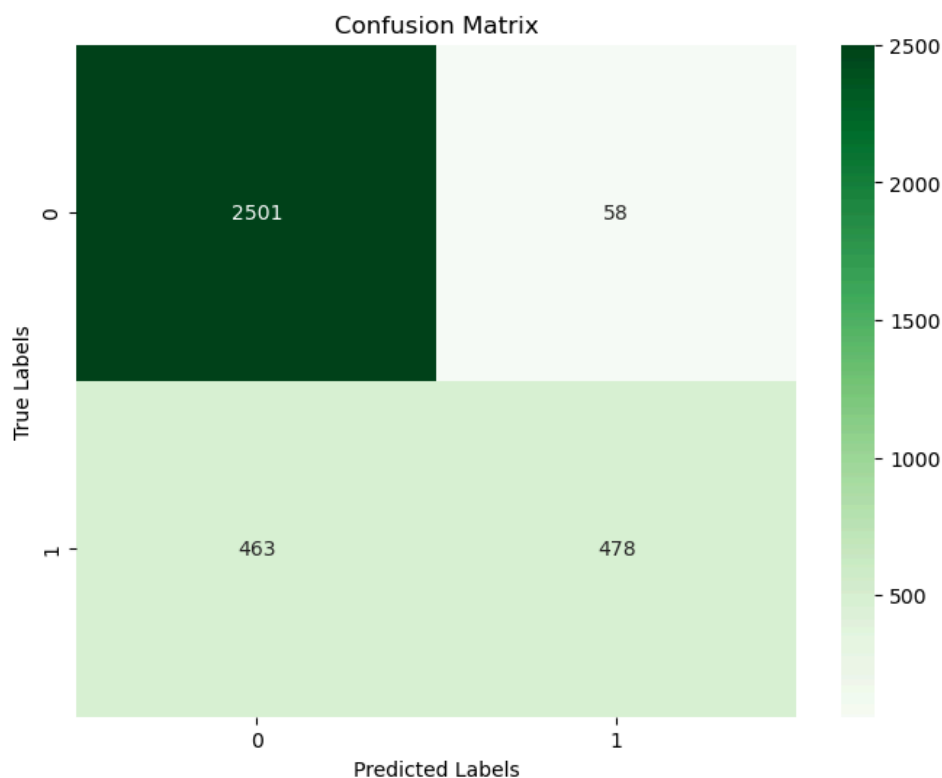
A técnica Random Forest, como pode ser visto na Figura 31, apresentou resultado minimamente melhor de acurácia, um pouco pior de precisão, e MCC basicamente igual. Deste modo, não se justifica o seu uso em vez de uma Árvore de Decisão para esses dados.

**Figura 33:** Aplicação do Gradient Boosting nos mesmo dados

```
# Gradient Boost classifier usage:
train_classifier3(xgb_clf, X_train_selected, y_train_selected)

Accuracy: 0.8511428571428571
Precision: 0.8917910447761194
Matthews Correlation Coefficient (MCC): 0.59748
```

**Figura 34:** Matriz de confusão de um Gradient Boosting



O método de Gradient Boosting, como visto na Figura 33, apresentou resultados piores que o da DT nas três métricas para avaliação que estão sendo computadas, indicando que pode não ser o mais apropriado para essa tarefa, uma vez que um modelo interpretável é interessante e apresenta resultados similares ou melhores.

O trabalho tinha também como um dos objetivos iniciais de estudar confiabilidade, análise de sobrevivência e manutenção preditiva, como foi feito nos capítulos anteriores. No entanto, não foi possível aplicar os conhecimentos de forma prática, desenvolvendo uma

estimativa do tempo de vida útil restante (RUL), para considerar a expectativa de vida do disco e para definir se um disco irá falhar em uma janela de tempo específica ou não. Seja para aplicá-lo junto aos atributos SMART para servir de *feature* que seria usado para treinar a árvore de decisão ou para utilizá-lo como um objeto de estudo.

O modelo desenvolvido, até o presente momento, é capaz de determinar, com uma alta acurácia, precisão e MCC, se um disco rígido ST4000DM000, dadas algumas de suas principais métricas SMART, falhou ou não. Dessa forma, é de grande interesse incrementar o modelo com conceitos de manutenção preditiva e RUL para torná-lo mais efetivo para prever falhas no futuro e sugerir possíveis manutenções e estimar a necessidade e a quantidade de compras de novos HDs, por exemplo.

## Capítulo 7. Conclusão

Este trabalho procurou tratar diversas áreas relacionadas ao tema de armazenamento de dados em nuvem. Para isso, incluiu desde o uso de HDDs, engenharia de confiabilidade, questões organizacionais em *data centers*, redundância, métricas para monitoramento de discos, conjunto de dados coletados de diversos discos em produção na Backblaze ao longo de anos, manutenção preditiva, aprendizado de máquina, entre outras. Foi então possível, chegar em um melhor entendimento geral do cenário, estudar um experimento que buscava prever falhas em discos com o uso de um algoritmo interpretável de aprendizado de máquina e replicá-lo em um modelo de disco diferente e em um período de tempo também diferente.

Dessa forma, além dos resultados práticos obtidos, também foi possível entender verdadeiramente a utilidade e os pontos fortes da maioria dos assuntos tratados. No caso do estudo dos dados da saúde dos discos usados na Backblaze, os principais pontos de interesse identificados são que o conjunto de dados usado serve como uma ferramenta de ensino para análise de dados, engenharia de confiabilidade e é ótimo para a aplicação de técnicas de IA e AM aplicadas a dados de discos rígidos do mundo real. Isso ocorre porque o dataset sobre as métricas usadas e monitoradas por HDs contém dados suficientes para criar correlações entre os atributos mais indicadores de falha e a falha de um disco em uma população. Os resultados são variados, mas a conclusão geral é que, embora seja possível prever a falha do disco até certo ponto, os resultados parecem estar limitados a um determinado modelo do disco. Deste modo, boa parte dos pontos estudados até o momento são concretizados.

Na área de Manutenção Preditiva, é possível ver que esta é uma área de estudo em que vale o investimento e aplicação de estratégias em produção. De acordo com Deloitte Insights [34], PdM é capaz de prometer uma redução de 20 a 50% no tempo necessário para planejar a manutenção, um aumento de 10 a 20% no tempo de atividade e na disponibilidade do equipamento e uma redução de 5 a 10% no custo geral de manutenção.

No caso dos resultados obtidos deste trabalho, ainda não é possível estimar de forma exata os valores de eficiência operacional que a utilização de uma estratégia de predição de falha nos discos por meio de um modelo de Árvore de Decisão como esse pode trazer para um data center ou para uma empresa de nuvem. No entanto, fica claro que pelo menos no caso de discos que compõem boa parte da infraestrutura, como era o caso do HD Seagate de 4TB nos anos estudados para a Backblaze, é um esforço que vale a pena pensando em aspectos monetários, para fazer manutenções conforme o necessário, e pensando nas garantias que a nuvem busca fornecer, pois permite aumentar a disponibilidade e durabilidade do sistema.

Muitos textos da literatura fazem referência à “vida útil remanescente” (RUL) como uma métrica fundamental para aplicações de manutenção preditiva, a qual oferece uma estimativa realista da vida útil restante de um componente até que ele tenha que ser completamente substituído. No entanto, apesar de sua utilidade, o conceito de RUL pode ser um conceito falho em alguns casos. Primeiro, a manutenção tradicional é organizada com base em suposições estatísticas de mecanismos de falha, principalmente relacionados ao desgaste normal, do tipo “*wear and tear*” ou deterioração por uso. Entretanto, de acordo com o ARC Advisory Group, [42] 82% das falhas de componentes são causadas por erros no processo, e não por problemas relacionados ao desgaste ou à idade.

Falhas por desgaste e uso podem sim ser previstas, porém consistem em apenas 18% das avarias (eventos de tempo de inatividade) [43]. Nos resultados, vê-se claramente uma curva de banheira, com a função de densidade de probabilidade (FDP) aumentando perto do “fim da vida útil”. Supondo que haja dados suficientes disponíveis, é também bem possível calcular com bastante precisão o tempo médio entre falhas (MTBF), o tempo médio para reparo (MTTR) e a vida útil restante (RUL).

Por esse motivo, é essencial levar em conta os eventos classificados como Cisnes Cinzas e Cisnes Pretos [15], que são, respectivamente, os eventos potencialmente imprevisíveis na forma de algo raro que você espera que vá ocorrer ou imprevisíveis que você nem espera que ocorra. Um exemplo prático de um evento Cisne Cinza poderia ser de 3 discos específicos que contêm todas as réplicas de um dado falhar em um período menor que o tempo médio de reparo (MTTR) do primeiro. Esse evento é extremamente improvável, mas pode ocorrer e deve ser contabilizado ao calcular a durabilidade do sistema de armazenamento. Uma vez que a escala é grande o suficiente, acontecimentos que seriam muito raros tornam-se comuns. Pensando na analogia de um dardo caindo de um avião em um campo com muitos alvos, a chance de cair no centro de um alvo específico é muito baixa, mas a chance de cair no centro de qualquer um dos alvos é bem mais alta. Felizmente, a probabilidade de eventos assim ocorrerem pode ser modelada ao utilizar as métricas corretas.

Para garantir a maior durabilidade, acima de tudo, é preciso que o sistema de armazenamento nunca fique sem espaço. Para isso, em um *data center*, é interessante que sempre existam canais de compra e reparo de HDs abertos, preferencialmente de antemão ao evento de falha ou antes da necessidade chegar. Dessa maneira, é interessante manter relações estreitas com os fabricantes de disco para garantir que o fornecimento da produção não seja interrompido. Como dito anteriormente, os processos de recebimento, manuseio, instalação e manutenção dos discos é bastante relevante pois é neles que podem estar localizadas muitas das falhas de equipamentos. Por esse motivo, é de grande interesse melhorar constantemente os processos para detectar problemas com antecedência, discos

potencialmente defeituosos não devem entrar em produção, e, como está sendo visto por todo este trabalho, o monitoramento dos discos funcionais e potencial de predição deles antes que ocorra a falha é muito relevante. Além disso, o monitoramento permite verificar o desempenho, taxas de falha anuais (AFR) e custo de cada modelo de disco, tornando possível tomar decisões mais adequadas para compra ou substituição gradual dos modelos de discos que não apresentam bons resultados, visto que esses últimos pontos são questões bastante relevantes que motivaram este trabalho ou tornaram-se claros conforme foi realizado o estudo.

Existe uma demanda real para identificar e substituir os discos problemáticos antes que eles realmente falhem. Quando um disco começa a falhar ou é identificado para substituição, é interessante que a equipe sempre tente restaurar o máximo de dados possível, pois isso garante que existam múltiplas opções para manter a durabilidade dos dados. Ponto esse que, juntamente com a disponibilidade (*uptime* do serviço), confiabilidade (qualidade do serviço) e capacidade de escalabilidade (*scale-out*) devem ser sempre maximizados na etapa de construir uma infraestrutura para servir uma plataforma provedora da nuvem (IaaS).

Para dar suporte a tudo isso, existe a redundância provida pela arquitetura e software, arquitetura Vault no caso da Backblaze, que garante que os dados permaneçam intactos mesmo que os discos individuais falhem. Existem níveis ainda acima de redundância presentes em todas as grandes nuvens públicas, em que racks inteiros, salas inteiras ou até *data centers* inteiros podem passar por falhas e perda de dados no nível local, mas isso não irá levar a uma perda total do dado ou falha de todo o sistema. Na maioria dos casos, parte de alguns sistemas fica fora do ar por no máximo alguns poucos minutos ao ano. Às vezes, esse *downtime* pode ser feito de forma proposital pelo provedor por ele busca atingir um nível ótimo de durabilidade [31], que seja extremamente seguro mas também seja viável de manter considerando o preço e devido aos diversos documentos, como Acordo de Nível de Serviço (SLAs) e Objetivos de Nível de Serviço (SLOs).

Por fim, todas as opções devem ser consideradas para atender às metas de escalabilidade, durabilidade e custo para manter o sistema. Dessa forma, uma estruturação e um entendimento dos principais pontos por trás do armazenamento de dados da nuvem é extremamente relevante para a nuvem apresentar sucesso. Pontos esses como o funcionamento dos componentes (HDDs e equipamentos que os suportam), estratégias de redundância dos dados (Replicação e *Erasure Coding*) e monitoramento do sistema para criar e aplicar técnicas, como modelos interpretáveis de AM, de predição e prevenção de falhas dos discos, evitando assim ainda mais as perdas de dados e downtime no serviço.

Com relação aos resultados obtidos, esses mostraram-se comparáveis com técnicas usadas em outros modelos de disco e podem ser usados como uma ferramenta útil para aplicar em produção para HDs desse tipo (Seagate de 4TB). Mais especificamente, a abordagem de Árvores de Decisão apresentou resultados muito bons, melhor ou igual aos modelos de Random Forest e Gradient Boosting que foram implementados no mesmo conjunto de dados. Além de que as DTs têm a significativa vantagem de serem interpretáveis, podendo produzir boas visualizações de como o modelo funciona, quais são os atributos mais relevantes e dando uma ideia embasada de como correlacionar métricas SMART aos discos.

Para trabalhos futuros, seria de grande interesse pensar em *pipelines* ou alternativas mais automatizadas para realizar estudos desse tipo que apliquem modelos de AM para diversos modelos de discos diferentes em um *data center* e que facilitem etapas como tratamento de dados e foco nas métricas mais importantes para correlacionar falha de HDs diferentes. Também pode ser relevante testar uma técnica comum na manutenção preditiva que não parece ser muito utilizada nos casos de HDs, que é a de geração de dados sintéticos de falhas. Esse processo pode ocorrer a partir do uso de modelagens de discos em uma simulação para que o sistema receba condições e atributos que levem-no a falhas ou defeitos de uma forma mais econômica e mais segura, com o objetivo final de tornar o conjunto de dados menos desbalanceado. Por fim, pode ser interessante também buscar integrar essas soluções, dos modelos preditivos para Discos Rígidos, na prática para empresas utilizarem esse conhecimento para melhorar a infraestrutura, o serviço provido e também se planejar de forma adequada em relação a compra de HDs e momento ideal para realizar manutenções.

## Agradecimentos

Agradeço imensamente a LuizaLabs, que apoia o projeto de extensão "**Pesquisa e desenvolvimento em tecnologias para data centers utilizando virtualização**" do DC-UFSCar, ao qual esta pesquisa de TCC está relacionada e teve como grande fonte de inspiração e aprendizado.

Agradeço também aos meus professores, amigos e familiares que tornaram tudo o que foi feito até então possível e melhor.

## Referências

Todas as Referências tiveram o último Acesso em 12 set. 2024.

- [1] - AMRAM, Maxime. et al. Interpretable predictive maintenance for hard drives. **Machine Learning with Applications**, v. 5, 2021. Disponível em: <https://doi.org/10.1016/j.mlwa.2021.100042>
- [2] - Ted Aravinthan. From Monolith to 'Modern' : Data Centers Today. In: LinkedIn. Disponível em: <https://www.linkedin.com/pulse/from-monolith-modern-data-centers-today-ted-aravinthan/>
- [3] - Molly Clancy. What's the Diff: Hot and Cold Data Storage. In: Backblaze. Disponível em: <https://www.backblaze.com/blog/whats-the-diff-hot-and-cold-data-storage/>
- [4] - Priya Dialani. The Future of Data Revolution will be Unstructured Data Disponível em: <https://www.analyticsinsight.net/insights/the-future-of-data-revolution-will-be-unstructured-data>
- [5] - Charles, Peter. et al. How Much Information? 2003. Disponível em: <https://www2.sims.berkeley.edu/research/projects/how-much-info-2003/index-2.html>
- [6] - Self-Monitoring, Analysis and Reporting Technology, In Wikipedia. Disponível em: [https://en.wikipedia.org/wiki/Self-Monitoring,\\_Analysis\\_and\\_Reporting\\_Technology](https://en.wikipedia.org/wiki/Self-Monitoring,_Analysis_and_Reporting_Technology)
- [7] - Backblaze. Hard Drive Test Data. Disponível em: <https://www.backblaze.com/cloud-storage/resources/hard-drive-test-data>.
- [8] - Steve Morgan. The World Will Store 200 Zettabytes Of Data By 2025. Disponível em: <https://cybersecurityventures.com/the-world-will-store-200-zettabytes-of-data-by-2025/>
- [9] - Backblaze. The Helium Factor and Hard Drive Failure Rates. Disponível em: <https://www.backblaze.com/blog/helium-filled-hard-drive-failure-rates/>
- [10] - TALEB, Nassim Nicholas. Antifragile: things that gain from disorder. New York : Random House, 2012.
- [11] - Erasure Code, In Wikipedia. Disponível em: [https://en.wikipedia.org/wiki/Erasure\\_code](https://en.wikipedia.org/wiki/Erasure_code)
- [12] - Machine Learning, In Wikipedia. Disponível em: [https://en.wikipedia.org/wiki/Machine\\_learning](https://en.wikipedia.org/wiki/Machine_learning)
- [13] - High Availability, In Wikipedia. Disponível em: [https://en.wikipedia.org/wiki/High\\_availability](https://en.wikipedia.org/wiki/High_availability)
- [14] - Backblaze. Managing for Hard Drive Failure and Data Corruption. Disponível em: <https://www.backblaze.com/blog/managing-for-hard-drive-failures-data-corruption/>
- [15] - TALEB, Nassim Nicholas. The Black Swan: The Impact of the Highly Improbable. Harlow, England: Penguin Books, 2008.
- [16] - Site reliability engineering, In Wikipedia. Disponível em: [https://en.wikipedia.org/wiki/Site\\_reliability\\_engineering](https://en.wikipedia.org/wiki/Site_reliability_engineering)
- [17] - Backblaze. 10 Stories From 10 Years of Drive Stats Data. Disponível em: <https://www.backblaze.com/blog/10-stories-from-10-years-of-drive-stats-data/>

- [18] - Backblaze. How Backblaze Scales Our Storage Cloud. Disponível em: <https://www.backblaze.com/blog/how-backblaze-scales-our-storage-cloud/>
- [19] - PINHEIRO, Eduardo. et al. Failure Trends in a Large Disk Drive Population. Google Inc., USENIX, 2007. Disponível em: [http://usenix.org/event/fast07/tech/full\\_papers/pinheiro/pinheiro.pdf](http://usenix.org/event/fast07/tech/full_papers/pinheiro/pinheiro.pdf)
- [20] - Acronis Drive Monitor: Disk Health Calculation. Disponível em: [https://care.acronis.com/s/article/9264-Acronis-Drive-Monitor-Disk-Health-Calculation?language=en\\_US](https://care.acronis.com/s/article/9264-Acronis-Drive-Monitor-Disk-Health-Calculation?language=en_US)
- [21] - Backblaze. What SMART Stats Tell Us About Hard Drives. Disponível em: <https://www.backblaze.com/blog/what-smart-stats-indicate-hard-drive-failures/>
- [22] - Seagate SMART Attribute Specification, 2011. Disponível em: <https://t1.daumcdn.net/brunch/service/user/axm/file/zRYOdwPu3OMoKYmBOby1fEEQEbU.pdf>
- [23] - Backblaze. How Long Do Disk Drives Last? Disponível em: <https://www.backblaze.com/blog/how-long-do-disk-drives-last/>
- [24] - Analysis of Hard Drive Reliability Trends: A 2023 Report by Backblaze. Disponível em: <https://www.guru3d.com/story/analysis-of-hard-drive-reliability-trends-a-2023-report-by-backblaze/>
- [25] - Predictive Maintenance & Monitoring using Machine Learning: Demo & Case study. 2018. 46 min. Publicado pelo canal Google Workspace. Disponível em: <https://www.youtube.com/watch?v=4wneZDEB3VA>
- [26] - Cisco. The Zettabyte Era Officially Begins (How Much is That?). Disponível em: <https://blogs.cisco.com/sp/the-zettabyte-era-officially-begins-how-much-is-that>
- [27] - Programa DB Browser for SQLite. Acesso e download gratuito disponível em: <https://sqlitebrowser.org/>
- [28] - SMOTE. Técnica de sobreamostragem minoritária sintética. Disponível em: [https://imbalanced-learn.org/stable/references/generated/imblearn.over\\_sampling.SMOTE.html](https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html)
- [29] - Autoencoders For Feature Selection. Disponível em: <https://hex.tech/blog/autoencoders-for-feature-selection/>
- [30] - R. Pinciroli. et al. Machine Learning Models for SSD and HDD Reliability Prediction. **Annual Reliability and Maintainability Symposium (RAMS)**. 2022. Disponível em: <https://ieeexplore.ieee.org/document/9893942>
- [31] - Intelligent Technical Solutions. Cloud Downtime Explained. Disponível em: <https://www.itsasap.com/blog/cloud-downtime-explained>
- [32] - Altexsoft. Predictive Maintenance: Employing IIoT and Machine Learning to Prevent Equipment Failures. Disponível em: <https://www.altexsoft.com/blog/predictive-maintenance/>
- [33] - Marshall Institute. Preventive Maintenance - The Cost of Maintaining Equipment. Disponível em: <https://insights.marshallinstitute.com/preventive-maintenance-the-cost-of-maintaining-equipment/>
- [34] - Deloitte Insights. Why predictive maintenance? Disponível em:

<https://www2.deloitte.com/content/dam/Deloitte/us/Documents/process-and-operations/us-predictive-maintenance.pdf>

[35] - Estimating Remaining Useful Life (RUL). 2019. 9m34s. Publicado por Matlab. Disponível em: [Estimating Remaining Useful Life \(RUL\) | Predictive Maintenance](#)

[36] - Ten quick tips for machine learning in computational biology. Disponível em: <https://biodatamining.biomedcentral.com/articles/10.1186/s13040-017-0155-3>

[38] - Backblaze. Hard Drive Stats for Q3 2018: Less is More. Disponível em: <https://www.backblaze.com/blog/2018-hard-drive-failure-rates/>

[39] - Survivorship Bias, In Wikipedia. Disponível em: [https://en.wikipedia.org/wiki/Survivorship\\_bias](https://en.wikipedia.org/wiki/Survivorship_bias)

[40] - Builtin. Correlation Matrix. Disponível em: <https://builtin.com/data-science/correlation-matrix>

[41] - Matriz de Confusão, in Wikipedia. Disponível em: [https://pt.wikipedia.org/wiki/Matriz\\_de\\_confus%C3%A3o](https://pt.wikipedia.org/wiki/Matriz_de_confus%C3%A3o)

[42] - Deloitte. Making maintenance smarter. Disponível em: <https://www2.deloitte.com/us/en/insights/focus/industry-4-0/using-predictive-technologies-for-asset-maintenance.html>

[43] - Aspen Tech. Remaining Useful Life. Disponível em: <https://www.aspentech.com/en/resources/blog/remaining-useful-life>

[44] - R. J. T. Morris and B. J. Truskowski, "The evolution of storage systems," in IBM Systems Journal, vol. 42, no. 2, pp. 205-217, 2003. Disponível em doi: 10.1147/sj.422.0205.

# Apêndices

## Códigos: tratamento do dataset

```

Python
# Combinação de todos os arquivos csv de cada trimestre em dataframes
# Posteriormente serão usados para criar um banco de dados
# cada dataframe irá se tornar uma tabela

import sqlite3
import pandas as pd
import os
import matplotlib.pyplot as plt
import numpy as np

cwd = os.getcwd() # Get no diretório atual

# Diretório contendo os arquivos CSV (absolute path)
csv_directory1 = os.path.join(cwd, 'Old_Disks/2018/2018q1')
...

# Lista para segurar os dataframes
dataframes1 = []
dataframes2 = []
dataframes3 = []
dataframes4 = []

columns = ['date', 'serial_number', 'model', 'capacity_bytes', 'failure',
'smart_1_normalized', 'smart_1_raw', 'smart_3_normalized', 'smart_3_raw',
'smart_5_normalized', 'smart_5_raw', 'smart_7_normalized', 'smart_7_raw',
'smart_10_normalized', 'smart_10_raw', 'smart_184_normalized',
'smart_184_raw', 'smart_187_normalized', 'smart_187_raw',
'smart_188_normalized', 'smart_188_raw', 'smart_190_normalized',
'smart_190_raw', 'smart_197_normalized', 'smart_197_raw',
'smart_198_normalized', 'smart_198_raw']

# Processamento de cada arquivo CSV
for csv_file in os.listdir(csv_directory1):
    if csv_file.endswith('.csv'):
        file_path = os.path.join(csv_directory1, csv_file)

        # Leitura do arquivo CSV para o Dataframe
        df2018q1 = pd.read_csv(file_path, usecols=columns)

        # Append do DataFrame na lista
        dataframes1.append(df2018q1)

```

```
# Mesma coisa é feita para os outros três trimestre

# Concatenação dos Dataframes em um só
combined_df2018q1 = pd.concat(dataframes1, ignore_index=True)

# Mesma coisa é feita para os outros trimestres
```

Python

```
# Transforma os dataframes de cada trimestre em tabelas diferentes dentro do
arquivo 20??disk.db

# Após esse passo será criado uma view que pega apenas as colunas relevantes
dos discos do modelo ST4000DM000

import sqlite3

# SQLite database name
db_name = '2018disk.db'

# Conecta ao SQLite database (Arquivo de banco de dados é criado caso já não
exista)
conn = sqlite3.connect(db_name)

# Insere o Combined DataFrameInsert em uma tabela
combined_df2018q1.to_sql('2018q1_table', conn, if_exists='replace',
index=False)

#Mesma coisa é feita para os outros trimestres do ano

# Mudanças são aplicadas e a conexão é fechada
conn.commit()
conn.close()
```

## Notebook Python

### 1. Análise Exploratória dos Dados (EDA), Exemplo 2016:

```
[2]: # Get the current working directory
    cwd = os.getcwd()

    # Directory containing CSV files (use absolute path)
    csv_directory_2016 = os.path.join(cwd, 'ModelSeagate4tb_2016.csv')

    ST4tb_disks_2016 = pd.read_csv(csv_directory_2016)

[3]: # Concatenação será usada para os diferentes anos
    file = pd.concat([ST4tb_disks_2016],axis = 0)

[4]: file0 = pd.concat([ST4tb_disks_2016],axis = 0)

[5]: # Tratamento dos Nulls
    pd.set_option('display.max_rows',None)
    ST4tb_disks_2016.isna().sum()

[5]: date                0
    serial_number        0
    failure              0
    smart_3_normalized   23
    smart_3_raw          23
    smart_5_normalized   23
    smart_5_raw          23
    smart_7_normalized   23
    smart_7_raw          23
    smart_187_normalized 23
    smart_187_raw        23
    smart_188_normalized 23
    smart_188_raw        23
    smart_190_normalized 23
    smart_190_raw        23
    smart_197_normalized 23
    smart_197_raw        23
    smart_198_normalized 23
    smart_198_raw        23
    dtype: int64

[6]: drop_columns = file.columns[file.isna().sum() > 400]
    file = file.drop(drop_columns, axis=1)

[7]: file.dropna(inplace =True)

[8]: file.isna().sum()
```

Subamostragem dos dados da classe 0, ou seja, dos discos sem falha.

## Sampling (Undersampling da classe 0)

```
from imblearn.under_sampling import RandomUnderSampler
from imblearn.over_sampling import SMOTE
from imblearn.pipeline import Pipeline

# Separate the features (X) and target variable (y)
X = file.drop('failure', axis=1)
y = file['failure']

# Defining the desired total number of samples after downsampling
desired_samples = 10000 # Adjust this between 5000 to 20000

# Calculate the number of samples to keep for class 1 (minority class)
class_1_count = sum(y == 1)
class_0_count = desired_samples - class_1_count

# Defining the sampling strategy for the majority class
sampling_strategy = {0: class_0_count}

# Using RandomUnderSampler to undersample the majority class while preserving the minority class
rus = RandomUnderSampler(sampling_strategy=sampling_strategy, random_state=42)
X_resampled, y_resampled = rus.fit_resample(X, y)

# Concatenate X_resampled and y_resampled to get the downsampled dataset
downsampled_df = pd.concat([pd.DataFrame(X_resampled), pd.DataFrame(y_resampled, columns = ['failure'])], axis=1)
```

Código do Modelo de DT, do sklearn, que está sendo treinado e utilizado.

```

from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, precision_score, matthews_corrcoef, confusion_matrix
import pandas as pd
from sklearn import metrics
from sklearn.tree import export_text, plot_tree
import matplotlib.pyplot as plt
dtc3 = DecisionTreeClassifier(max_depth=3)
dtc5 = DecisionTreeClassifier(max_depth=5)

conf = None
def train_classifier(dtc, X_train, y_train, X_test, y_test):
    dtc.fit(X_train, y_train)
    y_pred = dtc.predict(X_test)

    accuracy = accuracy_score(y_test, y_pred)
    print("Accuracy:", accuracy)

    precision = precision_score(y_test, y_pred)
    print("Precision:", precision)

    # Calculate the Matthews Correlation Coefficient
    mcc = matthews_corrcoef(y_test, y_pred)
    print(f'Matthews Correlation Coefficient (MCC): {mcc:.5f}')

    conf = confusion_matrix(y_test, y_pred)

    print(conf)
    # Display the confusion matrix using Seaborn heatmap for better visualization
    plt.figure(figsize=(8,6))
    sns.heatmap(conf, annot=True, fmt='d', cmap='Reds', xticklabels=True, yticklabels=True)
    plt.xlabel('Predicted Labels')
    plt.ylabel('True Labels')
    plt.title('Confusion Matrix')
    plt.show()

# Árvore de Decisão 'DT': dtc
train_classifier(dtc5, X_train_selected, y_train_re.values, X_test_selected, y_test.values)

```

## Códigos: métricas de avaliação do modelo

Python

```
# Acurácia
accuracy = accuracy_score(y_test,y_pred)
print("Accuracy:", accuracy)

# Precisão
precision = precision_score(y_test,y_pred)
print("Precision:", precision)

# MCC
mcc = matthews_corrcoef(y_test, y_pred)
print(f'Matthews Correlation Coefficient (MCC): {mcc:.5f}')

# Matriz de Confusão
conf = confusion_matrix(y_test, y_pred)
print(conf)
plt.figure(figsize=(8,6))
sns.heatmap(conf, annot=True, fmt='d', cmap='Reds')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.title('Confusion Matrix')
plt.show()
```