Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Bacharelado em Ciência da Computação

Previsão da Frequência de Acesso de Objetos em Serviços de Armazenamento em Nuvem Através de Algoritmos de Regressão

Matheus Franklin Rodrigues Silva

JUIZ DE FORA
JULHO, 2023

Previsão da Frequência de Acesso de Objetos em Serviços de Armazenamento em Nuvem Através de Algoritmos de Regressão

Matheus Franklin Rodrigues Silva

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Ciência da Computação

Orientador: Saulo Moraes Villela

Coorientador: Heder Soares Bernadino

JUIZ DE FORA

JULHO, 2023

Previsão da Frequência de Acesso de Objetos em Serviços de Armazenamento em Nuvem Através de Algoritmos de Regressão

Matheus Franklin Rodrigues Silva

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Saulo Moraes Villela
D.Sc. em Engenharia de Sistemas e Computação

Heder Soares Bernadino
D.Sc. em Modelagem Computacional

Alex Borges Vieira
D.Sc. em Ciências da Computação

Edelberto Franco Silva
D.Sc. em Computação

Carlos Cristiano Hasenclever Borges
D.Sc. em Engenharia Civil

JUIZ DE FORA

12 DE JULHO, 2023

Aos meus amigos, familiares e minha irmã.

Aos meus pais, pelo apoio incondicional.

Resumo

Os serviços de armazenamento em nuvem têm se tornado cada vez mais populares entre usuários domésticos e empresariais, devido às vantagens que oferecem, como serviço sob demanda, escalabilidade e não necessidade de compra e gerenciamento de infraestrutura própria de armazenamento físico. Os provedores desses serviços oferecem armazenamento hierárquico com diferentes preços baseados no nível de armazenamento utilizado. O objetivo deste trabalho é propor um modelo preditivo de classes para objetos armazenados em nuvem, que possibilite aos usuários escolherem de forma correta a classe de seus objetos com base nos seus padrões de acesso a dados. Para isso, o trabalho se dedica a explorar técnicas de aprendizado de máquina focadas em regressão que possam prever acessos futuros com base em padrões de acesso aos objetos, e através de um modelo de custo, classificar esses objetos corretamente. Será também avaliado o desempenho deste modelo ao se utilizar de bases reais de traços de dados de um serviço de armazenamento em nuvem, para que os usuários possam aproveitar de uma maior economia de custos, sem perda de qualidade e desempenho dos serviços de armazenamento.

Palavras-chave: Armazenamento em nuvem, Modelo preditivo, Padrões de acesso, Aprendizado de máquina, Regressão, Economia de custos.

Abstract

Cloud storage services have become increasingly popular among both home and business users due to the advantages they offer, such as on-demand service, scalability, and the need to purchase and manage physical storage infrastructure. Providers of these services offer hierarchical storage with different prices based on the level of storage used. The goal of this work is to propose a predictive class model for objects stored in the cloud, which enables users to correctly choose the class of their objects based on their data access patterns.To achieve this, the work focuses on exploring machine learning techniques focused on regression that can predict future accesses based on object access patterns, and through a cost model, correctly classify these objects. The performance of this model will also be evaluated using real trace data bases from a cloud storage service, so that users can take advantage of greater cost savings without sacrificing the quality and performance of storage services.

Keywords: Cloud storage, Predictive model, Access patterns, Machine learning, Regression, Cost saving.

Agradecimentos

Aos meus pais, pelo encorajamento, apoio e paciência. Sem o amor e apoio deles, eu não teria alcançado esse marco importante em minha vida.

Gostaria também de estender meus agradecimentos aos professores do Departamento de Ciência da Computação por seus ensinamentos valiosos e orientação durante todo o curso. Seus conhecimentos e dedicação foram essenciais para o meu crescimento acadêmico e profissional.

Agradeço também à minha namorada Isabela e aos meus amigos por seu constante incentivo e apoio. Suas palavras de encorajamento, compreensão e presença ao longo deste período foram essenciais para superar desafios e alcançar nossos objetivos.

“ Viver é melhor do que sonhar”.

Belchior (Como Nossos Pais)

Conteúdo

Lista de Figuras **8**

Lista de Tabelas **9**

Lista de Abreviações **10**

1 Introdução **11**
1.1 Apresentação do Tema . . . 11
1.2 Contextualização . . . 11
1.3 Descrição do Problema . . . 12
1.4 Motivação . . . 13
1.5 Objetivos . . . 13
1.6 Organização . . . 14

2 Fundamentação Teórica **16**
2.1 Armazenamento em Nuvem . . . 16
2.2 Aprendizado de Máquina . . . 18
2.2.1 *Decision Tree* . . . 19
2.2.2 *K-Nearest Neighbors* . . . 21
2.2.3 *Linear Regression* . . . 21
2.2.4 Lasso . . . 22
2.2.5 Regressão de Vetores Suporte . . . 22
2.3 Métricas de Avaliação . . . 24
2.3.1 Modelos de Regressão . . . 24
2.3.2 Modelos de Classificação Binária . . . 26
2.3.3 Economia de Custos . . . 28
2.4 Considerações Finais . . . 28

3 Trabalhos Relacionados **30**
3.1 Impacto do compartilhamento de conteúdo no consumo de largura de banda do armazenamento em nuvem . . . 30
3.2 Algoritmos online com base na frequência de acessos do objeto . . . 31
3.3 Modelo preditivo com base em padrões de acesso do usuário . . . 33
3.4 Considerações Finais . . . 34

4 Metodologia **35**
4.1 Modelo de Custo . . . 35
4.2 Modelo de Previsão . . . 36
4.3 Janelamento de Tempo . . . 37
4.4 Configuração do Modelo . . . 39

5 Experimentos e Resultados **42**
5.1 Bases de Dados . . . 42
5.2 Métricas . . . 43
5.3 Resultados . . . 44

6 Conclusão **57**
6.1 Trabalhos Futuros 58

Bibliografia **59**

Lista de Figuras

2.1 Exemplo de um classificador utilizando árvore de decisão. 20
2.2 Ilustração de um exemplos simples de SVR. 24

4.1 Limite do número de acessos $\hat{y}_i = 1$ para as classes frequente (em azul) e infrequente (em vermelho), considerando os preços da Tabela 4.1. 36
4.2 Janela deslizante de treino e previsão em um único objeto onde período e o tamanho do passo são de quatro semana. 39
4.3 Janela deslizante de treino e previsão em um único objeto onde período é de quatro semanas e o tamanho do passo é de uma semana. 39

Lista de Tabelas

4.1 Principais componentes da estrutura de preços adotada pela maioria dos provedores de armazenamento em nuvem hierárquico: exemplo dos servidores do Iowa do Google Cloud Storage para três camadas com base nas classes dos objetos (frequente, infrequente e inativo). Preços de referência junho 2023. 36

5.1 Resultados das métricas nos teste na base PoP-1 com o janelamento 1 × 1. 45
5.2 Resultados das métricas nos teste na base PoP-1 com o janelamento 4 × 1. 46
5.3 Resultados das métricas nos teste na base PoP-1 com o janelamento 4 × 4. 47
5.4 Resultados das métricas nos teste na base PoP-2 com o janelamento 1 × 1. 48
5.5 Resultados das métricas nos teste na base PoP-2 com o janelamento 4 × 1. 49
5.6 Resultados das métricas nos teste na base PoP-2 com o janelamento 4 × 4. 50
5.7 Resultados dos custos nos teste na base PoP-1 com o janelamento 1 × 1. . 51
5.8 Resultados dos custos nos teste na base PoP-1 com o janelamento 4 × 1. . 52
5.9 Resultados dos custos nos teste na base PoP-1 com o janelamento 4 × 4. . 53
5.10 Resultados dos custos nos teste na base PoP-2 com o janelamento 1 × 1. . 54
5.11 Resultados dos custos nos teste na base PoP-2 com o janelamento 4 × 1. . 55
5.12 Resultados dos custos nos teste na base PoP-2 com o janelamento 4 × 4. . 56

Lista de Abreviações

DCC	Departamento de Ciência da Computução
UFJF	Universidade Federal de Juiz de Fora
IA	Inteligência Artificial
GB	*Gigabyte*

1 Introdução

1.1 Apresentação do Tema

Os serviços de armazenamento em nuvem têm atraído usuários domésticos e empresariais ao redor do mundo devido a suas vantagens como serviço fornecido sob demanda, a não necessidade de compra e gerenciamento de uma infraestrutura própria de armazenamento físico de dados, agilidade e escalabilidade. Além do mais, os provedores desses serviços oferecem armazenamento em nuvem em camadas com várias opções de preços com base em classes de frequência de acessos a objetos (HSU et al., 2018).

O tema deste trabalho consiste em examinar um aspecto relacionado aos custos desses serviço para os usuários: prever a quantidade de acessos a determinados objetos, definir um modelo de cálculo de valor limite capaz de classificá-los como frequentes ou infrequentes e a partir de então atribuí-los ao nível de armazenamento apropriado, com a finalidade de reduzir os custos do usuário ao mesmo tempo em que este possa usufruir de um acesso mais veloz para seus arquivos ativos. Assim sendo, este trabalho se dedica a propor um modelo de aprendizado de máquina capaz de prever as classes apropriadas com base em padrões de acesso a dados. E por fim, se concentra em avaliar o desempenho desse modelo ao se utilizar de bases reais orientadas a traços de dados de um serviço de armazenamento em nuvem.

1.2 Contextualização

Nos últimos anos, é possível testemunhar uma grande inversão dos métodos de armazenamento de dados utilizados por ambos usuários domésticos e corporativos. Métodos tradicionais que consistem no uso de dispositivos locais como discos rígidos, mídias óticas, pendrives e data centers para armazenamento de objetos digitais estão sendo substituídos por serviços em nuvem. O mais recente índice global de computação em nuvem da Cisco prevê um aumento de 4,6 vezes do total de volume de dados armazenados na nuvem entre

2016 à 2022 (CISCO, 2019).

Fica evidente também o crescimento da adesão à nuvem de pequenas e médias empresas, de forma que o cenário pandêmico vivido recentemente demanda uma rápida adaptação para garantimento de uma operação estável da grande maioria dos setores em modelo de trabalho remoto, essas empresas então fazem uso das vantagens oferecidas por esse tipo de serviço tal qual backups, replicação de dados em diferentes locais, compartilhamento de dados e trabalho colaborativo.

Hoje em dia, essas empresas voltam seus esforços para tentar extrair o máximo de eficiência do uso desses serviços, seja através da obtenção de um menor tempo de acesso a seus arquivos armazenados e principalmente através do barateamento dos custos de armazenamento.

1.3 Descrição do Problema

Os provedores de serviços oferecem armazenamento em nuvem em camadas com várias opções de preços com base em classes de frequência de acessos a objeto (HSU et al., 2018). Normalmente, arquivo de dados criados recentemente são acessados com mais frequência por usuários destes provedores. Ao longo do tempo, os acessos ao objeto diminuem gradativamente, tornando-os infrequentes.

Assim, é comum que os provedores desses serviços adotem camadas de armazenamento com um acesso rápido, mas de preço maior por volume para objetos acessados frequentemente e também camadas, cujo o preço por volume é menor e portanto o acesso ao objeto seja mais lento, para objeto infrequentes. Existindo camadas com preços ainda menores e latência de acesso maiores, para objetos inativos.

Apesar do comportamento típico descrito acima, há objetos ativos que variam seus acessos entre frequentes e infrequentes. Usuários do serviço de armazenamento em nuvem devem alocar seus objetos nas classes apropriadas, de acordo com seus padrões de acesso recentes, e fazê-lo com a maior urgência a fim de ter um serviço adequado e reduzir seus custos.

1.4 Motivação

Otimizações destinadas aos serviços de armazenamento em nuvem fundamentados em acesso a dados têm provocado pesquisas da indústria e da academia. É comum encontrar estes esforços de pesquisa focalizados em otimizar a infraestrutura de nuvem dos provedores. Porém, raros são os trabalhos que exploram possibilidades de um melhor desempenho destinado ao usuário final.

Facultado o desenvolvimento dessa infraestrutura com destino à divisão do armazenamento em camadas de acordo com o nível de frequência de acesso a dados, as pesquisas mais recentes exploram métodos para que os usuários escolham de forma correta a classe de seus objetos. Esses métodos têm focados em algoritmos online, que tomam decisões em tempo real, assim que novos dados estão disponíveis, rastreando os acessos a cada objeto individualmente para definir sua camada, sem examinar uma previsão de classes baseada nos padrões de acesso de vários objetos em conjunto (LIU; PAN; LIU, 2019; LIU; PAN; LIU, 2021; ERRADI; MANSOURI, 2020).

Nesse sentido, este trabalho se determina a propor um modelo preditivo de classes para objetos armazenados em nuvem, através de um algoritmo offline que utiliza de técnicas de aprendizado de máquina para explorar padrões de acessos de um conjunto de objetos.

1.5 Objetivos

O objetivo geral deste trabalho é propor um modelo preditivo para que os usuários sejam capazes de prever a quantidade de acessos futuros de objetos que ainda estão ativos e atribuí-los adequadamente nas camadas de arquivo frequentes ou não frequentes de provedores de serviços de armazenamento em nuvem. Esse modelo será baseado em técnicas de aprendizado de máquina que exploram padrões de acesso a objetos. Os objetivos específicos incluem: desenvolver um arcabouço para tratamento de dados, aplicações de técnicas de aprendizado de máquina para prever acessos futuros a objetos, criar um modelo de custos capaz de calcular um valor limite de acessos para determinar a classe dos objetos, realizar simulações orientadas ao rastreamento de visitas de usuários a objetos

em serviços de armazenamento reais e por fim avaliar o desempenho do modelo proposto e compará-lo com outros métodos presentes na literatura.

O desafio desta pesquisa é criar uma estrutura para previsões precisas e rápidas utilizando os acessos mais recentes de objetos. Desta forma, ao passo que os objetos estejam ativos, os usuários podem aproveitar acesso de baixa latência das camadas de armazenamento frequentes e infrequentes oferecidas pelos provedores e ao mesmo tempo economizar custos.

1.6 Organização

O presente trabalho está estruturado em seis capítulos, que abordam diferentes aspectos do trabalho de pesquisa.

No Capítulo 1 é apresentado uma introdução ao tema, destacando sua importância e relevância. São estabelecidos os objetivos do estudo e uma visão geral da estrutura do trabalho.

O Capítulo 2 é dedicado à fundamentação teórica, onde são descritos os termos, conceitos e técnicas fundamentais que foram utilizados no desenvolvimento do trabalho. São apresentados os fundamentos teóricos necessários para compreender os métodos e abordagens empregados ao longo do estudo.

No Capítulo 3 são apresentados os trabalhos relacionados. Nessa seção, são discutidos trabalhos acadêmicos e pesquisas prévias que ofereceram contribuições relevantes para o tema abordado no presente trabalho. É realizada uma revisão da literatura, explorando estudos anteriores que tratam de questões semelhantes ou relacionadas ao problema em questão.

O Capítulo 4 descreve a metodologia adotada neste trabalho. São detalhados os procedimentos e as técnicas utilizadas para o desenvolvimento do modelo proposto. São apresentadas as etapas de implementação e experimentação, fornecendo uma visão clara dos passos seguidos durante o estudo.

No Capítulo 5 são apresentados os resultados da pesquisa e a avaliação do modelo proposto. São analisados e discutidos os dados coletados, juntamente com os resultados das métricas de avaliação utilizadas. São destacados os pontos fortes e limitações do

modelo.

Por fim, no Capítulo 6, são apresentadas as conclusões finais do trabalho. São sumarizados os principais resultados alcançados, as contribuições do estudo e as considerações finais. São destacadas as implicações práticas e teóricas do trabalho, bem como sugestões para futuras pesquisas relacionadas ao tema.

2 Fundamentação Teórica

Neste capítulo são descritos os termos e técnicas empregadas no desenvolvimento do trabalho e aborda três tópicos principais: armazenamento em nuvem, aprendizado de máquina e métricas de avaliação. O primeiro tópico fornece uma visão geral do armazenamento de arquivos em serviços de nuvem, incluindo sua definição, funcionamento, vantagens de sua utilização e os principais provedores desse tipo de serviço no mundo. O segundo tópico se concentra no conceito de aprendizado de máquina destacando os diferentes tipos de aprendizado, com ênfase no aprendizado de máquina supervisionado. Além disso, é apresentado o conceito de algoritmos de regressão e explorado diversos destes algoritmos, incluindo o *K-Nearest Neighbors*, *Decision Tree*, *Support Vector Regression*, Lasso e Regressão Linear. Por fim, o terceiro tópico aborda as métricas de avaliação utilizadas para mensurar o desempenho dos modelos de regressão, tais como o coeficiente de determinação (R^2), erro médio absoluto (MAE), erro quadrático médio (MSE) e raiz do erro quadrático médio (RMSE).

2.1 Armazenamento em Nuvem

O armazenamento em nuvem é um modelo de computação em nuvem que permite aos usuário finais armazenarem seus dados e arquivos na internet fazendo uso de um provedor de serviços em nuvem acessado através de uma rede pública ou privada. É função do provedor gerenciar, armazenar e garantir a segurança de servidores, infraestrutura e rede, além de assegurar o acesso aos dados a qualquer momento, com capacidade praticamente ilimitada e flexível. Esse tipo de modelo não requer a compra e o gerenciamento de uma infraestrutura própria de armazenamento, assim como fornece flexibilidade, capacidade escalar, acessibilidade global e confiabilidade para acesso aos dados.

Presentemente, é possível encontrar no mercado diversas opções de provedores de armazenamento na nuvem. Os mais populares são *Simple Storage Service* (também chamado de S3, da *Amazon*), *Google Drive*, *Dropbox*, *Microsoft OneDrive* e *iCloud*. To-

dos eles oferecem a capacidade de armazenar, acessar e compartilhar dados e arquivos, facilitando a colaboração entre pessoas, gerenciamento de dados, implantação de serviços e o aumento da agilidade. Com o foco em uma melhor relação custo-benefício, também é comum que esses provedores ofereçam diferentes configurações de performance e retenção para o uso do armazenamento. Os dados acessados com pouca frequência podem ser armazenados em camadas de armazenamento de custo mais baixo, porém de maior latência de acesso, ao passo que existem opções de camadas com custo mais alto e menor latência para dados acessados frequentemente. Há também, opções de armazenamento para dados inativos, com custo ínfimo de armazenamento, porém custo maior de acesso, usualmente utilizadas para *backup*.

Os principais provedores de serviços de armazenamento em nuvem utilizam três componentes fundamentais no cálculo do custo de armazenamento: volume (em *bytes*), operações (número de acessos) e requisições (*bytes* por acesso). Com base nessa estrutura de preço Ribeiro et al. (2020) propõem um modelo de custos, apresentado na Equação (2.1), que representa o custo que se tem por período para armazenar um objeto i na classe j em um serviço de armazenamento em nuvem. Nesta equação, x_i e y_i representam respectivamente o volume e o número de acessos que o objeto i recebe por período. O custo de armazenamento de um objeto na classe j é calculado levando em consideração o custo de armazenamento por GB v_j, o custo por operação de acesso o_j e o custo pela requisição de acesso r_j (ou seja, o total de volume de acesso em GB). Nesse modelo, o período de armazenamento é considerado, mas foi omitido na formulação para facilitar a compreensão. Para simplificar, foi atribuído custos iguais nas operações de leitura e gravação, mas o modelo pode ser facilmente adaptado para representar custos distintos para cada uma delas.

$$C_{i_j}(x_i, y_i) = v_j \times x_i + o_j \times y_i + r_j \times x_i \times y_i. \tag{2.1}$$

Através da Equação (2.1) é estimado um limite do número de acessos em que o custo de armazenamento seja igual para duas classes diferentes. Por tanto, sendo $\hat{y_i}$ este limite, basta igualar os custos para as classes distintas a e b por $C_{i_a}(x_i, \hat{y_i}) = C_{i_b}(x_i, \hat{y_i})$. A Equação (2.2) representa este limite de acessos, onde os parâmetros custo por volume

(v), custo por operação (o) e custo por requisição (r) são definidos pelos provedores do serviço de armazenamento.

$$\hat{y_i} = \frac{x_i \times (v_a - v_b)}{(o_a - o_b) + x_i \times (r_b - r_a)}. \tag{2.2}$$

A Equação (2.2) oferece duas percepções interessantes em relação ao volume do objeto. Primeiramente, ao aumentar o volume dos objetos, observa-se que o limite de acessos tende a ser determinado pela relação entre os custos por volume e custos por requisição $\frac{v_a - v_b}{r_b - r_a}$. Em segundo lugar, ao diminuir o volume dos objetos, o limite de acessos tende a zero, indicando que a classe frequente é sempre a opção de menor custo para objetos muito pequenos. Em resumo, o modelo de custo fornece o limite de acessos necessário para classificar as classes de objetos com base nos preços adotados pelo provedor de infraestrutura em nuvem, permitindo assim classificar um objeto através do valor de acesso predito.

2.2 Aprendizado de Máquina

Aprendizado de Máquina é uma área de IA cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado bem como a construção de sistemas capazes de adquirir conhecimento de forma automática. Um sistema de aprendizado é um programa de computador que toma decisões baseado em experiências acumuladas através da solução bem sucedida de problemas anteriores (MONARD; BARANAUSKAS, 2003).

O Aprendizado de Máquina examina e estuda a estruturação e desenvolvimento de algoritmos capazes de aprender a partir de seus erros e prever características sobre dados. Estes algoritmos agem através da generalização a partir da entrada de dados amostrais com a finalidade de fazer previsões ou tomar decisões com base nesses padrões identificados, ao invés de seguir instruções estáticas programadas.

Existem três principais tipos de aprendizado de máquina: supervisionado, não supervisionado e por reforço. Neste trabalho será dada ênfase ao aprendizado de máquina supervisionado. O aprendizado supervisionado ocorre quando a partir de um conjunto de entrada e saída o modelo é capaz de aprender uma função capaz de mapear qualquer

entrada para uma saída correta. Norvig e Russel (2002) definem matematicamente a tarefa do aprendizado supervisionado como: Dado um conjunto de treinamento com N exemplos de pares de entrada e saída, representado na Equação (2.3), onde cada y_j é gerado por uma função desconhecida $y = f(x)$, deseja-se encontrar uma função h, capaz de se aproximar da função real f. Neste exemplo x e y podem ser de quaisquer valores, não precisam necessariamente de serem números. A função h é chamada de hipótese. O aprendizado é definido como a procura através do espaço de possíveis hipóteses por uma que performe bem, mesmo com novos conjuntos de exemplos fora do conjunto de treinamento. Para medir-se a precisão de uma hipótese é dado um conjunto de exemplos distintos do conjunto de treinamento, denominado conjunto de teste. É dito que uma hipótese tem boa generalização se esta prediz corretamente o valor de y para os novos exemplos.

$$Z = \{(x_1, y_1), \ldots, (x_n, y_n)\}. \tag{2.3}$$

Os algoritmos de regressão, como parte do aprendizado de máquina supervisionado, desempenham um papel crucial na modelagem da relação entre variáveis independentes (também chamadas de características ou atributos) e uma variável dependente contínua. Eles se destacam especialmente na previsão da frequência de acesso, pois vão além da simples classificação de objetos em categorias pré-definidas, como “alto”, “médio” ou “baixo” acesso. Em vez disso, esses algoritmos fornecem estimativas numéricas da frequência de acesso, permitindo uma previsão mais precisa e detalhada. Isso possibilita compreender o comportamento dos objetos armazenados em nuvem de forma mais granular, considerando as nuances da frequência de acesso em diferentes cenários.

As subseções seguintes descrevem os algoritmos de regressão utilizados.

2.2.1 *Decision Tree*

Árvore de decisão (*decision tree*) é uma técnica de aprendizado supervisionado que constrói uma estrutura em forma de árvore para modelar a relação entre as variáveis independentes e a variável dependente. Essa estrutura é composta por decisões em forma de nós internos e consequências em forma de folhas.

Uma árvore de decisão atinge sua decisão ao executar uma sequência de testes.

Cada nó interno na árvore corresponde a um teste do valor de um dos atributos de entrada, A_i, e os ramos dos nós são rotulados com os possíveis valores do atributo, $A_i = v_{ik}$. Cada nó folha na árvore especifica um valor a ser retornada pela função (NORVIG; RUSSEL, 2002).

A Figura 2.1 apresenta a representação de uma árvore de decisão simples para o problema de decidir quando esperar por uma mesa em um restaurante. Neste exemplo, os nós são representados pelos atributos "Clientes no restaurante?" e "Estimativa de espera", onde os seus valores são testados nos ramos pertencentes a cada nó, por fim, as folhas indicam as classes associadas a cada decisão.

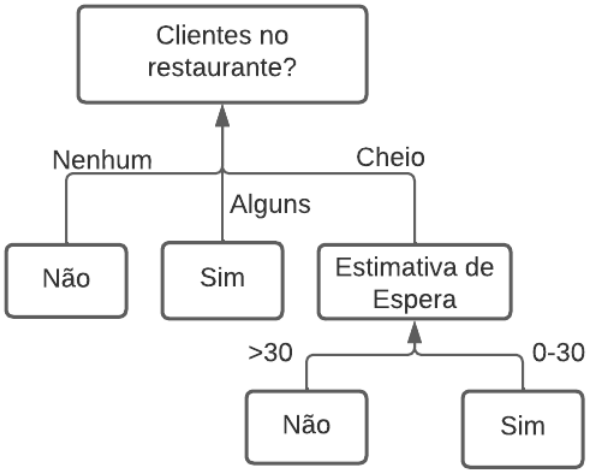

Figura 2.1: Exemplo de um classificador utilizando árvore de decisão.

A diferença entre os algoritmos usados para regressão e classificação reside, em primeiro lugar, na natureza da variável dependente. Enquanto na regressão essa variável é contínua, na classificação ela é discreta, representando classes ou categorias. Além disso, a escolha da função de custo para avaliar a qualidade das divisões nos nós da árvore também difere entre os dois casos. Na classificação, métricas como entropia ou índice de Gini são usadas para medir a impureza dos dados em relação às classes, visando maximizar a separação e pureza das categorias. Por outro lado, na regressão, métricas como a redução do erro quadrático médio ou a minimização da variância são comumente empregadas para encontrar a melhor divisão dos dados, visando a minimização dos erros de previsão e a obtenção de estimativas mais precisas.

2.2.2 *K-Nearest Neighbors*

O algoritmo *K-Nearest Neighbors* (KNN) é um dos mais famosos algoritmos de classificação usados para predição de classes de um registro com classe não especificada baseado na classe dos registros vizinhos. No geral, para predizer a classe de uma nova amostra o algoritmo olha para amostras similares dentre o conjunto de treino, logo se um registro tem n atributos, então este será considerado com um vetor no espaço n-dimensional e sua classe sera prevista como a classe do vizinho mais próximo, com base em um critério de distância neste espaço, por exemplos distância euclidiana (KUHKAN, 2016).

O funcionamento do algoritmo para regressão é relativamente simples. Uma vez selecionados os k vizinhos mais próximos, o algoritmo calcula uma média ou ponderação dos valores da variável dependente desses vizinhos para obter a previsão final. No caso da média simples, a previsão será a média dos valores da variável dependente dos k vizinhos. Se ponderações forem utilizadas, os vizinhos mais próximos podem ter maior influência na previsão, dependendo da sua proximidade em relação ao objeto de entrada.

2.2.3 *Linear Regression*

Regressão linear (*linear regression*) é um método estatístico utilizado para modelar a relação entre uma variável dependente contínua e uma ou mais variáveis independentes. Ele assume uma relação linear entre essas variáveis, onde a variável dependente é estimada como uma combinação linear ponderada das variáveis independentes, juntamente com um termo de erro.

Uma função linear univariável (uma reta) com entrada x e saída y é da forma $y = w_1x + w_0$, onde w_0 e w_1 são coeficientes com valores reais a serem aprendidos. É usado a letra w para dar a ideia de coeficientes como pesos (*weigths*), o valor de y é alterado ao se mudar o peso relativo de um termo ou de outro. Define-se w como o vetor $\{w_0, w_1\}$, e define-se $h_w(x) = w_1x + w_0$. Chama-se de regressão linear a tarefa de achar h_w que melhor encaixa os dados de entrada e saída. Para encaixar essas dados em uma linha, o que deve ser feito é encontrar os valores de pesos $\{w_0, w_1\}$ que minimizem a perda empírica (NORVIG; RUSSEL, 2002).

A regressão linear univariável pode ser estendida para um problema multivariável

ao se aprender um vetor de pesos w de tamanho n, dado uma entrada x como sendo um vetor de n elementos.

2.2.4 Lasso

Lasso (*Least Absolute Shrinkage and Selection Operator*) é um modelo linear de regressão que estima coeficientes esparsos. Ele é útil em alguns contextos devido à sua tendência de preferir soluções com menos coeficientes diferentes de zero, reduzindo efetivamente o número de recursos dos quais a solução depende. Por esta razão, Lasso e suas variantes são fundamentais para o campo de sensoriamento comprimido. Sob certas condições, ele pode recuperar o conjunto exato de coeficientes diferentes de zero.

A regressão Lasso visa identificar as variáveis e coeficiente de regressão correspondentes que levam a um modelo que minimize o erro de predição. Isso é alcançado ao impor uma restrição nos parâmetros do modelo que força a soma dos valores absolutos dos coeficientes sejam menos que um valor fixo (λ), fazendo com que os coeficientes de regressão "encolham" à zero. De forma prática, isso restringe a complexidade do modelo. Variáveis com um coeficiente de regressão igual a zero após o encolhimento são excluídas. A escolha de λ é frenquentemente feita usando uma abordagem automatizada de validação cruzada *k-fold*. Para esta abordagem, o conjunto de dados é dividido aleatoriamente em k subamostras de tamanhos iguais. Enquanto as $k-1$ subamostras são usadas para desenvolver um modelo de previsão, a subamostra restante é usada para validar este modelo. Este procedimento é realizado k vezes, com cada uma das k subamostras. Um resultado geral é produzido ao combinar os k resultados de validação para um intervalo de λ valores e ao escolher o λ preferido, que é então usado para determinar o modelo final (RANSTAM; COOK, 2018).

2.2.5 Regressão de Vetores Suporte

As máquinas de vetores suporte (*support vector machines* - SVMs) vêm recebendo crescente atenção da comunidade de aprendizado de máquina nos últimos anos. Os resultados da aplicação dessa técnica são comparáveis e muitas vezes superiores aos obtidos por outros algoritmos populares de aprendizado, tal como RNAs (FACELI et al., 2011)). A

regressão de vetores suporte (*support vector regression* - SVR) é o nome dado para a aplicação de máquinas de vetores suportes ao problema de regressão.

A regressão de vetores suporte é um método que fornece flexibilidade para se definir qual o máximo de erro aceitável no método e a partir daí ela calcula uma área entre linhas (ou um hiperplano em conjuntos de treino com mais de um coeficiente) que engloba esses dados.

O algoritmo tem como objetivo encontrar uma função $h(x)$ que produza saídas contínuas para os dados de treinamento que desviem no máximo de ϵ de seu rótulo desejado. Essa função deve também ser o mais uniforme e regular possível (FACELI et al., 2011).

No caso do uso de funções lineares h, a regularidade se reflete em buscar uma função com pequeno w, o que pode ser conseguido pela minimização da norma $||w||$. Tem-se então o problema de otimização apresentado na Equação (2.4), com as restrições expressas na Equação (2.5).

$$\min \frac{1}{2}||w||^2 \tag{2.4}$$

$$\begin{cases} y_i - w \cdot x_i - b \leq \epsilon_i \\ w \cdot x_i + b - y_i \leq \epsilon_i \end{cases} \tag{2.5}$$

Procura-se então a função linear que aproxime os pares (x_i, y_i) de treinamento com uma precisão de ϵ (FACELI et al., 2011).

A Figura 2.2 apresenta um exemplo simples do procedimento realizado pelo algoritmo. O objetivo do algoritmo é encontrar a função linear tal que os dados de treinamento fiquem na região compreendida entre os erros.

Além disso, uma característica importante dos algoritmos de *Support Vector Regression* é a possibilidade de utilizar diferentes *kernels* para modelar os dados. O *kernel* é uma função que mapeia os dados de entrada para um espaço de maior dimensionalidade, permitindo encontrar um hiperplano que melhor separa as classes. Alguns dos *kernels* mais comumente utilizados são o linear, que realiza uma projeção linear dos dados; o sigmoide, que utiliza a função sigmoidal para mapear os dados; e o RBF (*Radial Basis Function*), que mapeia os dados para um espaço infinito de dimensionalidade. Cada

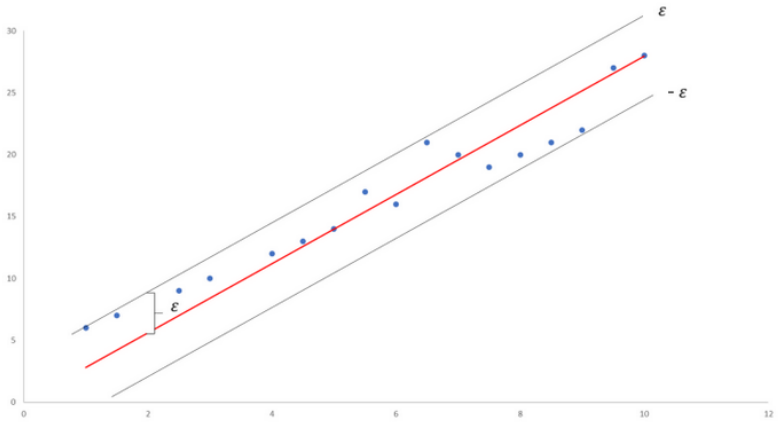

Figura 2.2: Ilustração de um exemplos simples de SVR.

kernel tem suas características e é mais adequado para determinados tipos de dados e problemas.

2.3 Métricas de Avaliação

As métricas de avaliação são ferramentas essenciais para avaliar o desempenho de modelos de aprendizado de máquina. Elas fornecem uma maneira objetiva de medir a qualidade das previsões e classificações feitas pelos modelos. Nesta seção, apresentaremos diferentes métricas de avaliação adequadas para modelos de regressão, modelos de classificação binária e métricas para avaliação de economia de custos.

2.3.1 Modelos de Regressão

As métricas de avaliação para modelos de regressão são utilizadas para medir a precisão das previsões feitas pelos modelos em relação aos valores reais. As principais métricas utilizadas incluem:

- **Erro Médio Absoluto (MAE)**

O erro médio absoluto (*mean absolute error* - MAE) calcula a média das diferenças absolutas entre as previsões e os valores reais. É uma medida simples de erro que indica o quão próximo as previsões estão dos valores reais.

Tem seu cálculo expresso na Equação (2.6), onde n é o número de amostras de

teste, y_i o valor real do i-ésimo exemplo e $\hat{y}_i$ o valor previsto pelo modelo para o i-ésimo exemplo. Quanto menor o valor de MAE melhor o desempenho.

$$MAE = \frac{1}{n}\sum |y_i - \hat{y}_i|. \tag{2.6}$$

- **Erro Quadrático Médio (MSE)**

O erro quadrático médio (*mean squared error* - MSE) calcula a média dos erros quadráticos entre as previsões e os valores reais. A fórmula para o MSE está expressa na Equação (2.7), onde os termos $n, y_i, \hat{y}_i$ têm o mesmo significado que na equação do MAE.

$$MSE = \frac{1}{n}\sum (y_i - \hat{y}_i)^2. \tag{2.7}$$

O MSE penaliza erros maiores de forma mais significativa do que o MAE, devido à operação de elevação ao quadrado. Quanto menor o valor do MSE, melhor é o desempenho do modelo.

- **Raiz Quadrada do Erro Quadrático Médio (RMSE)**

A raiz quadrada do erro quadrático médio (*root mean squared error* - RMSE) é a raiz quadrada do MSE e fornece uma medida do desvio padrão dos erros. É uma métrica comumente usada para representar a dispersão dos erros em relação aos valores reais. Quanto menor o valor do RMSE, melhor é o desempenho do modelo.

$$RMSE = \sqrt{MSE}. \tag{2.8}$$

- **Coeficiente de Determinação (R^2)**

O R^2 fornece uma medida da proporção da variabilidade dos valores de destino que é explicada pelo modelo. Varia entre 0 e 1, onde 0 indica que o modelo não é capaz de explicar a variabilidade dos dados e 1 indica que o modelo explica toda a variabilidade. A Equação (2.9) representa o cálculo do R^2, onde os termos $n, y_i, \hat{y}_i$ têm o mesmo significado que na equação do MAE e o termo $\overline{y}$ representa a média dos valores reais.

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \overline{y})^2}. \tag{2.9}$$

2.3.2 Modelos de Classificação Binária

Embora este trabalho utilize modelos de regressão para prever acessos futuros, é realizado posteriormente um processo de classificação binária com base no modelo de custos. Portanto, as métricas de avaliação para modelos de classificação binária podem ser aplicadas para avaliar o desempenho do trabalho.

Ao avaliar o desempenho de um modelo de classificação binária, quatro termos são frequentemente utilizados: Verdadeiro Positivo (VP), Verdadeiro Negativo (VN), Falso Positivo (FP) e Falso Negativo (FN). O Verdadeiro Positivo representa o número de instâncias positivas que foram corretamente classificadas como positivas pelo modelo. Por sua vez, o Verdadeiro Negativo indica o número de instâncias negativas que foram corretamente classificadas como negativas pelo modelo.

Os Falsos Positivos, por outro lado, são as instâncias negativas que foram incorretamente classificadas como positivas pelo modelo. Já os Falsos Negativos referem-se às instâncias positivas que foram erroneamente classificadas como negativas pelo modelo. As principais métricas para classificação binária incluem:

- **Acurácia** (Acc)

A acurácia mede a proporção de instâncias corretamente classificadas em relação ao total de instâncias, tem seu cálculo dado na Equação (2.10). Quanto maior a acurácia, melhor o desempenho do modelo.

$$\text{Acurácia} = \frac{(VP + VN)}{VP + VN + FP + FN}. \tag{2.10}$$

- **Precisão** (P)

A precisão representa a proporção de instâncias positivas corretamente classificadas em relação ao total de instâncias classificadas como positivas. É uma medida de quão precisas são as previsões positivas do modelo. É expressa na Equação (2.11).

$$P = \frac{VP}{VP + FP}. \tag{2.11}$$

- **Revocação** (R)

A revocação (*recall*) também conhecida como taxa de verdadeiros positivos, é a proporção de instâncias positivas corretamente classificadas em relação ao total de instâncias reais positivas. Mede a capacidade do modelo de identificar corretamente os casos positivos. Apresentada na Equação (2.12).

$$R = \frac{VP}{VP + FN}. \tag{2.12}$$

- F_1***-score***

O F_1-*score* é uma medida combinada da precisão e da revocação, calculada a partir da média harmônica entre essas duas métricas. É útil quando se deseja levar em consideração tanto a precisão quanto a revocação. Indicada na Equação (2.13).

$$F_1\text{-}score = \frac{2 \cdot P \cdot R}{P + R}. \tag{2.13}$$

- F_β**-score**

O F_β-score é uma extensão do F_1-score que permite ajustar o equilíbrio entre a precisão e a revocação, através do parâmetro beta. Um valor de beta < 1 dá mais peso à revocação, tornando o modelo mais sensível à identificação de exemplos positivos, enquanto um valor de beta > 1 dá mais peso à precisão, tornando o modelo mais conservador na classificação dos exemplos como positivos. Sua fórmula é apresentada na Equação (2.14).

$$F_\beta\text{-}score = \frac{(1 + \beta^2) \cdot R \cdot P}{\beta^2 \cdot R + P}. \tag{2.14}$$

- **Área sob Curva ROC**

A Curva ROC (*Receiver Operating Characteristic*) é uma curva que representa a taxa de verdadeiros positivos em função da taxa de falsos positivos em diferentes pontos de corte do modelo. A área sob a curva ROC (AUC-ROC) é uma métrica que resume o desempenho geral do modelo. Quanto maior a AUC-ROC, melhor o desempenho do modelo.

2.3.3 Economia de Custos

Uma métrica relevante para a avaliação do modelo proposto é a economia de custo relativa (*Relative Cost Saving*) proposta por Gonçalves et al. (2016). Essa métrica quantifica a economia relativa de custos alcançada pelo modelo em comparação com uma política de referência. Ela é calculada como a diferença percentual entre os custos totais obtidos com o modelo proposto e os custos totais obtidos com a política de referência, dividida pelos custos totais obtidos com a política de referência. A RCS permite uma avaliação direta do benefício econômico do modelo proposto em relação à política de referência, fornecendo informações valiosas sobre sua eficácia na redução de custos.

2.4 Considerações Finais

Neste capítulo foram abordados três tópicos principais que fornecem a fundamentação teórica necessária para o desenvolvimento deste trabalho. Inicialmente, foi apresentado o conceito de armazenamento em nuvem, descrevendo seu funcionamento, vantagens e os principais provedores desse tipo de serviço no mundo. A capacidade de escolher diferentes configurações de desempenho e retenção de armazenamento, oferecida pelos provedores, é crucial para este trabalho.

Em seguida, foi explorado o conceito de aprendizado de máquina, com ênfase no aprendizado de máquina supervisionado. O aprendizado de máquina permite a construção de sistemas capazes de adquirir conhecimento automaticamente, através do desenvolvimento de algoritmos que aprendem a partir de erros e são capazes de fazer previsões com base em padrões identificados nos dados. O aprendizado supervisionado, em particular, envolve o treinamento de um modelo utilizando um conjunto de dados de entrada e saída conhecidos, com o objetivo de aprender uma função capaz de mapear novos exemplos de entrada para saídas corretas.

Além disso, foram apresentados os algoritmos de regressão utilizados neste trabalho, *Decision Tree*, *K-Nearest Neighbors*, *Linear Regression*, Lasso e *Support Vector Regression*. Esses algoritmos desempenham um papel fundamental na modelagem da relação entre variáveis independentes e uma variável dependente contínua, permitindo a

previsão da frequência de acesso aos objetos armazenados em nuvem. Cada algoritmo possui características e abordagens diferentes, proporcionando opções para explorar a relação entre as variáveis e obter estimativas mais precisas e detalhadas.

Por fim, foram apresentadas as métricas de avaliação utilizadas para mensurar o desempenho do modelo proposto. No próximo capítulo são apresentados alguns trabalhos relacionados ao artigo desenvolvido.

3 Trabalhos Relacionados

Nesta etapa do desenvolvimento do trabalho foi realizada uma pesquisa por trabalhos acadêmicos que oferecessem contribuições relevantes para este estudo. O primeiro trabalho pesquisado abordou o impacto do compartilhamento de conteúdo no consumo de largura de banda do armazenamento em nuvem, e foi de extrema importância, uma vez que dele foi obtida a base de dados utilizada na avaliação deste trabalho. Em seguida, foram encontrados estudos relacionados a propostas de algoritmos de previsão da classe de frequência de objetos com base na frequência anterior de acessos, bem como um trabalho que tratava de modelos preditivos para a classe de frequência de acesso de objetos, levando em consideração os padrões de acessos do usuário.

Os trabalhos pesquisados servem de base para um melhor desenvolvimento deste trabalho. A seguir, são apresentados alguns trabalhos que mais se assemelham ou contribuem para o desenvolvimento de um modelo de previsão de frequência de acessos visando a otimização de custos aos usuários de serviços de armazenamento em nuvem.

3.1 Impacto do compartilhamento de conteúdo no consumo de largura de banda do armazenamento em nuvem

A utilização de serviços de armazenamento em nuvem tem se tornado cada vez mais comum, tanto para usuários domésticos quanto empresariais. No entanto, o aumento da popularidade desses serviços também tem gerado um grande impacto no consumo de largura de banda da internet. O compartilhamento de conteúdo entre dispositivos de um mesmo usuário pode resultar em múltiplos downloads do mesmo conteúdo, o que não apenas desperdiça largura de banda, mas também sobrecarrega os servidores de armazenamento.

Gonçalves et al. (2016) realizaram um estudo sobre o impacto do compartilha-

mento de conteúdo no consumo de largura de banda do armazenamento em nuvem, com foco no tráfego gerado pelo Dropbox. Eles coletaram dados de quatro redes diferentes e analisaram o tráfego para identificar a proporção de downloads relacionados ao compartilhamento de conteúdo entre dispositivos.

Os resultados obtidos pelos autores mostraram que uma grande parcela dos downloads realizados pelos usuários do Dropbox está associada ao conteúdo compartilhado entre vários dispositivos. Especificamente, eles observaram que cerca de 57% a 70% dos downloads são relacionados a conteúdos compartilhados. Com base nesses dados, os autores propuseram uma arquitetura alternativa de sincronização que utiliza caches para aliviar os servidores de armazenamento desses downloads repetitivos.

Os experimentos realizados pelos pesquisadores demonstraram que a abordagem proposta foi eficaz na redução significativa dos downloads repetitivos, proporcionando benefícios tanto para os provedores de serviços de armazenamento em nuvem, quanto para a rede e os usuários finais. A utilização de caches permitiu evitar a maioria dos downloads desnecessários, otimizando o consumo de largura de banda e melhorando o desempenho do serviço.

Este estudo foi de extrema importância para a realização deste trabalho, uma vez que os dados coletados durante essa pesquisa foram utilizados como base para a avaliação dos modelos propostos.

3.2 Algoritmos online com base na frequência de acessos do objeto

Alguns artigos propõem algoritmos capazes de guiar usuários nas tomadas de decisão sobre transferência de objetos entre camadas frias e quentes de serviços de armazenamento, a fim de otimizar os custos. Esses algoritmos levam em consideração a frequência de acessos anterior do objeto.

Esse é o caso de Liu, Pan e Liu (2019), que propõem um novo algoritmo online para auxiliar usuários de serviços de armazenamento em nuvem na decisão de transferir objetos entre as camadas quente e fria. Neste modelo de algoritmo é considerado apenas

objetos de dados somente leitura, que incluem como exemplo arquivos de vídeos e fotos compartilhados em redes sociais. Para um usuário de nuvem, cada um de seus objetos é representado por 3 características: $\{r(t), v(t), R(t)\}$, que representam respectivamente o número de requisições de leitura, o tamanho do objeto e dos dados recuperados, em um tempo t. O custo cobrado por cada objeto é composto por três partes: o custo incidido pelo armazenamento, pela leitura de dados e pela recuperação de dados, caso exista.

O custo de armazenamento para um objeto armazenado na classe fria ou na classe quente no tempo t, denotados por C_{S_c} e C_{S_h}, respectivamente e apresentados na Equação (3.1), sendo S_c e S_h, respectivamente, os custos para armazenamento na classe fria e na classe quente. Já o custo de acesso consiste do custo incidido pela pelas requisições de leitura e os dados recuperados de um objeto no tempo t, tomando C_{r_c} e C_{r_h} como custo de acesso na classe fria e na classe quente respectivamente, expressados na Equação (3.2).

$$\begin{aligned} C_{S_c}(t) &= S_c v(t), \\ C_{S_h}(t) &= S_h v(t). \end{aligned} \tag{3.1}$$

$$\begin{aligned} C_{r_c}(t) &= r(t) r_c + R(t) R_c, \\ C_{r_h}(t) &= r(t) r_h + R(t) R_h. \end{aligned} \tag{3.2}$$

Há também o custo de transferência entre classes, que consiste no custo de uma requisição de reescrita, dada por O_c na classe fria e O_h na classe quente, e no custo de recuperação de dados, representado por R_c para a classe fria e R_h para a quente. Com seu cálculo expressado na Equação (3.3).

$$\begin{aligned} C_t(t, t+1) &= O_c + v(t) R_c, \ frio \rightarrow quente \\ C'_t(t, t+1) &= O_h + v(t) R_h, \ quente \rightarrow frio. \end{aligned} \tag{3.3}$$

Para fim de otimização de custos, pode-se calcular quando o desconto de ler um objeto na camada quente é maior ou menor do que o desconto de se armazenar o mesmo objeto na camada fria, e então decidir sobre sua transferência de camadas. De toda forma, os usuários geralmente não sabem sobre a frequência de acessos futuros de seus objetos. Assim é proposto o algoritmo online, para apoiar essa decisão sem nenhum dado sobre

acessos futuros, que calcula um ponto de equilíbrio β_c e a partir da frequência de acessos acumulada para cada objeto, é analisado quando o objeto passa a receber mais ou menos leituras que β_c, classificando-o em quente ou frio.

Por fim, através de extensas simulações, Liu, Pan e Liu (2019) mostram que o algoritmo online proposto é capaz de garantir uma economia significativa de custos ao usuário comparado a sempre manter objetos na classe fria ou sempre transferir dados entre classes, quando sua frequência começa a variar.

Outro trabalho relevante é o de Erradi e Mansouri (2020), que propõe dois algoritmos online de otimização de custos para lidar com o mesmo problema. O primeiro algoritmo usa *No Replication* (NR), ou seja, não cria cópias do mesmo objeto. Este, inicialmente coloca os objetos na camada quente e, com base nas próximas solicitações de acesso de leitura ou escrita, pode decidir movê-los para a camada fria a fim de se otimizar o custo do serviço de armazenamento. E o segundo algoritmo *With Replication* (WR), que inicialmente coloca os objetos na camada fria e, em seguida, os replicam na camada quente ao receber solicitações de leitura ou escrita.

Também é mostrada a eficácia dos algoritmos propostos por meio de um extenso estudo de simulação usando uma base de dados de rastreamento de carga de trabalho real do Twitter e o simulador CloudSim, onde é confirmado uma economia de 5% a 55% comparado a políticas de não migração, que armazena permanentemente objetos na classe quente.

3.3 Modelo preditivo com base em padrões de acesso do usuário

Além dos algoritmos online, existem abordagens que utilizam modelos preditivos baseados em padrões de acesso do usuário para estimar a classe de frequência de acesso de objetos em serviços de armazenamento em nuvem.

Ribeiro et al. (2020) propuseram um arcabouço composto por três componentes - metadados, predição e controle - para prever a classe adequada de objetos em serviços de armazenamento em nuvem. O componente metadados coleta informações dos dados

gerenciados pelo serviço de armazenamento em nuvem e dos usuários e dispositivos que acessam esses dados. O componente de predição atua utilizando os metadados coletados para aprender características dos usuários e seus padrões de acesso, fazendo predições de acessos a objetos, a fim de otimizar a operação do serviço. E por fim, o componente controle atua como interface entre o serviço virtual de armazenamento e o provedor de infraestrutura de nuvem, com o trabalho de modificar ou não a classe dos objetos, quando o modelo de predição está devidamente treinado.

O arcabouço proposto foi avaliado utilizando traços reais de acessos de usuários do Dropbox, coletados por Gonçalves et al. (2016). Os resultados obtidos demonstraram um potencial significativo de economia de custos de armazenamento, com reduções de até 25%. Além disso, foram avaliados modelos de aprendizado de máquina, com foco em algoritmos de classificação binária, que alcançaram economias de custos de 3% e 15% em relação a manter todos os objetos na classe "frequente", para as bases de dados Pop-1 e Pop-2, respectivamente.

3.4 Considerações Finais

Neste capítulo são apresentados trabalhos acadêmicos que propõem soluções para o problema de fornecer ao usuário de serviços de armazenamento em nuvem ferramentas ou recursos capazes de auxiliá-lo na decisão de qual classe de frequência utilizar no armazenamento de seus objetos, a fim de se obter melhor economia de custos.

É possível perceber a evolução dos trabalhos, cada vez aprimorando os modelos utilizados por seus antecessores, porém todos usando uma abordagem de transformação de uma base de dados de séries temporais em uma base de dados distintas em duas classes. A partir do estudo desses artigos é criada uma base teórica mais sólida para o desenvolvimento da proposta deste trabalho.

4 Metodologia

Neste capítulo é apresentada a metodologia adotada neste trabalho, baseada na proposta de Ribeiro et al. (2020). No entanto, é adotada uma abordagem distinta, tratando o problema como um problema de séries temporais. Em vez de classificar os objetos previamente e, em seguida, prever a classe futura dos objetos, o modelo prevê diretamente a quantidade de acessos futuros de cada objeto e, em seguida, classifica os objetos de acordo com o modelo de custos.

Uma abordagem adicional deste trabalho é a utilização de diferentes janelamentos de tempo para avaliar o modelo. Essa abordagem envolve o deslocamento dos intervalos de tempo dos dados durante os testes, o que nos permite analisar os padrões de acesso em períodos diversos. Com isso, podemos avaliar a capacidade do modelo em prever o acesso futuro em diferentes intervalos de tempo e compreender como os padrões de acesso dos usuários contribuem para essa previsão.

Por fim, é apresentado uma seção de configuração do modelo, na qual descrevemos as configurações utilizadas para a avaliação do modelo. Isso inclui a escolha dos algoritmos de aprendizado de máquina, os parâmetros utilizados, as bibliotecas de software empregadas e outras considerações relevantes para a implementação e o desempenho do modelo.

4.1 Modelo de Custo

Na Tabela 4.1 são apresentados os componentes de custo de armazenamento no Google Cloud para os níveis de acesso frequente, infrequente e inativo. Este estudo tem como objetivo reduzir os custos de armazenamento sem comprometer a qualidade do serviço oferecido aos usuários, portanto tem foco nas classes de objetos frequentes e infrequentes. É importante destacar que os provedores líderes do mercado mantêm um alto padrão de qualidade para essas categorias, garantindo uma menor latência no acesso aos objetos.

Com base nessa estrutura de preço, foi utilizado o modelo de Ribeiro et al. (2020),

Tabela 4.1: Principais componentes da estrutura de preços adotada pela maioria dos provedores de armazenamento em nuvem hierárquico: exemplo dos servidores do Iowa do Google Cloud Storage para três camadas com base nas classes dos objetos (frequente, infrequente e inativo). Preços de referência junho 2023.

Componente	Frequente	Infrequente	Inativo
Volume (GB)	0.0200	0.010	0.004
Operação (1K acessos)	0.0004	0.001	0.010
Requisições (GB)	0.0000	0.010	0.020

apresentado no Capítulo 2. Na Figura 4.1, é ilustrado o limite de número de acessos para um objeto de volume igual a 1 GB, levando em consideração os preços indicados na Tabela 4.1. O limite estimado $\hat{y}_i$ é de 1 acesso. Quando o número de acessos de um objeto é inferior a 1 em um determinado intervalo de tempo, ele deve ser classificado como infrequente. Por outro lado, se o número de acessos for igual ou superior a 1, a opção de baixo custo é a classe frequente (camada padrão).

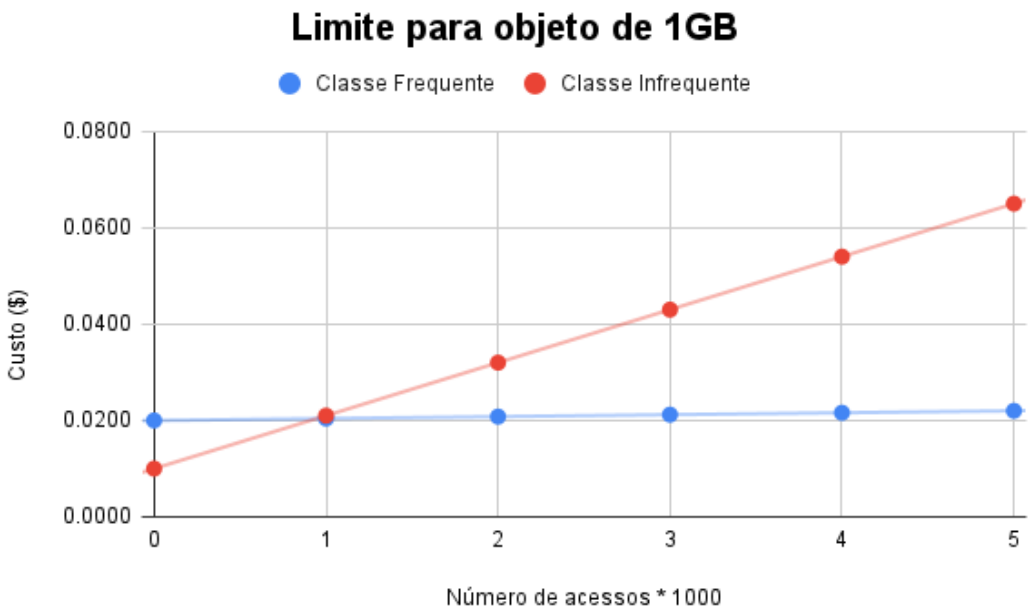

Figura 4.1: Limite do número de acessos $\hat{y}_i = 1$ para as classes frequente (em azul) e infrequente (em vermelho), considerando os preços da Tabela 4.1.

4.2 Modelo de Previsão

O objetivo principal do modelo de previsão é decidir a qual classe um objeto específico deve ser atribuído no próximo período de tempo, com base nos padrões de acesso do

usuário. Os dados utilizados pelo modelo constituem uma série temporal, que inclui o histórico de acessos ao longo do tempo e o tamanho do arquivo em GB em cada instante.

O pseudocódigo do Algoritmo 1 fornece uma visão geral da estrutura que visa prever classes de objetos com base nos padrões de acesso em um serviço de armazenamento. A função de previsão desempenha um papel fundamental no modelo, pois tem como objetivo identificar padrões nos dados de acesso e estimar a quantidade de acessos futuros. Para realizar essas previsões, o componente utiliza de um dos algoritmos de regressão descritos no Capítulo 2, selecionando aquele mais adequado para o contexto do problema.

Algoritmo 1: Modelo de previsão proposto.

Entrada: *dadosDeAcessos*; *limiteDeAcessos*; *modelo*;
Saída: *classes*;
1 **início**
2 $entradaTreino \leftarrow$ recuperaEntradaTreino(*dadosDeAcessos*);
3 $rotuloTreino \leftarrow$ recuperaRotuloTreino(*dadosDeAcessos*);
4 $entradaPrevisao \leftarrow$ recuperaEntradaPrevisão(*dadosDeAcessos*);
5 $modelo \leftarrow$ treinaModelo($entradaTreino$, $rotuloTreino$);
6 $previsaoDeAcessos \leftarrow$ preveAcessos(*modelo*, *entradaPrevisao*);
7 $classes \leftarrow$ classificaObjetos($previsaoDeAcessos$, $limiteDeAcessos$)
retorna *classes;*
8 **fim**

A função de classificação dos objetos determina a classe de um objeto no próximo período de tempo. Essa classificação é baseada na previsão de acessos para o período futuro, obtida previamente, e no limite de acessos calculado pelo modelo de custos descrito na seção anterior. Em particular, a classificação dos objetos é de extrema importância, uma vez que influencia diretamente os custos e a viabilidade do serviço de armazenamento.

4.3 Janelamento de Tempo

Para deslocar os intervalos de tempo dos dados de treinamento e teste dos algoritmos de previsão, foi adotada uma janela deslizante de comprimento fixo. Por simplificação, utilizamos o termo “período” para se referir ao tamanho da janela empregada no treinamento e no teste, e o termo “tamanho do passo” para indicar o deslocamento desses “períodos” a cada iteração da janela deslizante.

Em cada iteração da janela, o modelo é dividido em duas partes: treinamento e teste. No estágio de treinamento, o algoritmo de previsão recebe os dados de acesso de cada objeto durante um período específico, utilizando esses dados como variável independente. Além disso, são incluídos o somatório dos acessos de cada objeto no período seguinte, que serve como variável dependente. Isso permite que o algoritmo aprenda os padrões de acesso dos usuários. No estágio de teste, o algoritmo de previsão utiliza os dados de acesso do objeto no período que foi utilizado como variável dependente no treinamento, agora como variável independente, para prever o número de acessos no período seguinte.

Dessa forma, o modelo sempre utiliza os dados dos dois períodos anteriores ao período que se deseja prever como treinamento, enquanto o período anterior é utilizado como entrada para o teste.

A Figura 4.2 ilustra as janelas utilizadas em cada iteração para um objeto específico. Nessa representação, são apresentadas 24 semanas, divididas em 6 períodos. Na primeira linha, observamos que o primeiro período e o segundo período (cada um com duração de 4 semanas) são utilizados como entrada e variável dependente para treinamento do modelo, respectivamente. Além disso, o segundo e o terceiro períodos são utilizados como entrada e previsão do modelo. É importante ressaltar que o segundo período é empregado tanto como variável dependente do treinamento quanto como entrada do estágio de teste do modelo. Na segunda linha, é mostrada a próxima iteração da janela deslizante, na qual todas as janelas de tempo são deslocadas, em um tamanho do passo igual a 4 semanas, para o próximo período. Esse processo é repetido ao longo do tempo, de modo que cada previsão é baseada apenas no treinamento realizado no período anterior, sem considerar os dados anteriores. Esse padrão se mantém até que o último período seja usado como variável dependente da avaliação.

É importante ressaltar que a estrutura descrita anteriormente é adequada para a organização dos testes do modelo quando o tamanho do período é igual ao tamanho do passo. No entanto, quando o tamanho do passo é menor do que o tamanho do período, é necessário fazer algumas alterações. A Figura 4.3 ilustra como seria o janelamento nesse caso, utilizando um período de 4 semanas, mas um passo de apenas 1 semana.

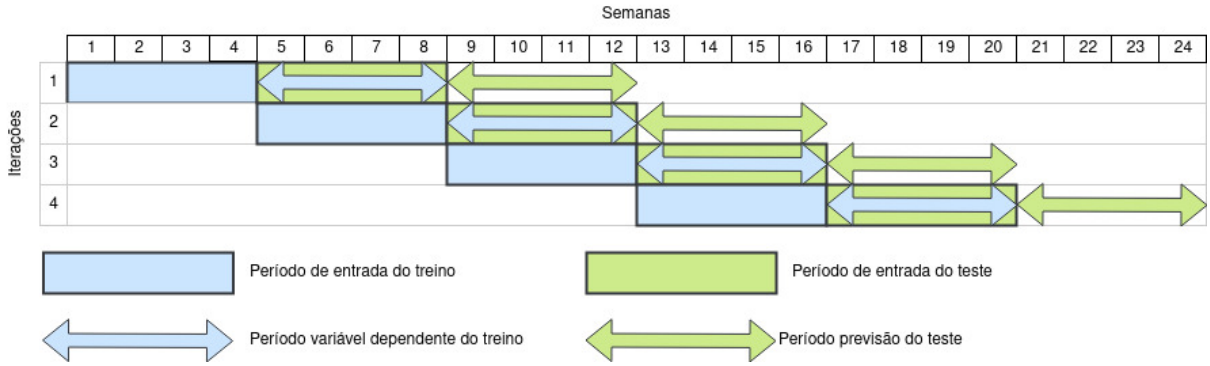

Figura 4.2: Janela deslizante de treino e previsão em um único objeto onde período e o tamanho do passo são de quatro semana.

Agora, é importante observar que tanto o período utilizado como variável dependente no treinamento do modelo quanto o período de previsão têm o mesmo tamanho do que o passo. Isso ocorre porque não faz sentido prever além desses períodos, uma vez que na próxima iteração esses períodos já serão previstos.

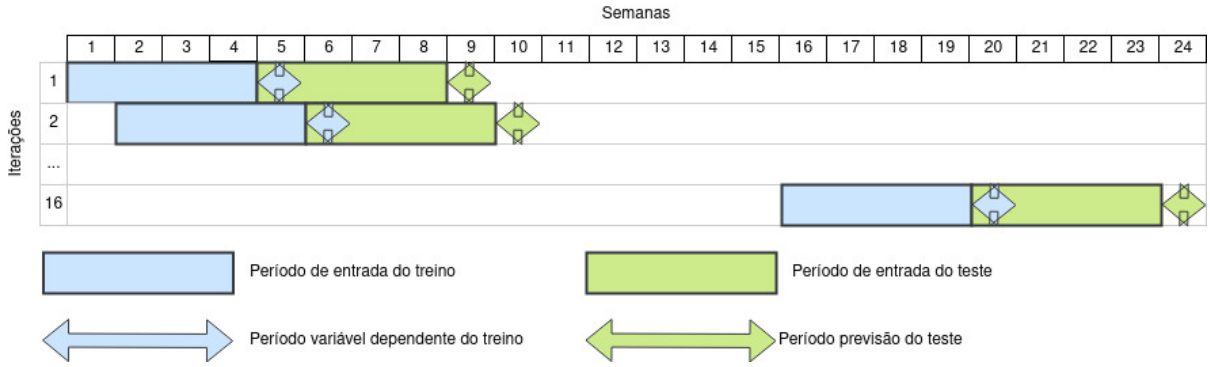

Figura 4.3: Janela deslizante de treino e previsão em um único objeto onde período é de quatro semanas e o tamanho do passo é de uma semana.

4.4 Configuração do Modelo

Na implementação do modelo, a linguagem de programação Python foi escolhida devido à sua ampla adoção e às bibliotecas especializadas em inteligência artificial e aprendizado de máquina. Durante o desenvolvimento do modelo, foram utilizadas as bibliotecas Pandas (TEAM, 2023), Scikit-Learn (PEDREGOSA et al., 2011) e NumPy (HARRIS et al., 2020). A biblioteca Pandas foi empregada para a manipulação e processamento dos dados, facilitando as tarefas de limpeza, transformação e agregação dos dados utilizados no treinamento e teste do modelo. A Scikit-Learn foi fundamental para a aplicação

dos algoritmos de aprendizado de máquina, proporcionando uma variedade de modelos e métodos para a construção e avaliação do modelo de previsão. Já o NumPy foi utilizada para operações eficientes de manipulação de matrizes e vetores.

No pseudocódigo do Algoritmo 2 são apresentadas as adaptações realizadas no modelo preditivo para permitir a realização das avaliações. Essas adaptações incluem o janelamento de tempo, que consiste em dividir a série temporal em janelas de tempo fixas, permitindo a análise dos padrões de acesso em períodos específicos. Além disso, o algoritmo incorpora o cálculo das métricas de avaliação, que são essenciais para a avaliação do desempenho do modelo. Por fim, também é adicionado uma etapa de filtragem de dados, que tem como objetivo remover os arquivos com volume zero na última semana de entrada dos períodos de treino e de teste, separadamente. Isso ocorre porque esses arquivos ainda não existem nessa janela de avaliação e, portanto, não podem ser considerados na previsão e classificação.

Algoritmo 2: Algoritmo de avaliação.

```
Entrada: dadosDeAcessos, dadosDeTamanho, limiteDeAcessos, modelo
Saída: dadosDeAvaliacao
1  início
2      dadosTreino ← recuperaDadosTreino(dadosDeAcessos);
3      dadosTeste ← recuperaDadosTeste(dadosDeAcessos);
4      enquanto dadosTeste ≤ dadosDeAcessos.tamanho faça
5          dadosTreino ← filtraDados(dadosDeTamanho);
6          dadosTeste ← filtraDados(dadosDeTamanho);
7          modelo ← treinaModelo(dadosTreino);
8          previsaoDeAcessos ← preveAcessos(modelo, dadosTeste);
9          classes ← classificaObjetos(previsaoDeAcessos, limiteDeAcessos);
10         dadosDeAvaliacao+ = avaliacao(classes, dadosTeste);
11         deslizaJanela(dadosTreino, dadosTeste)
12     fim enqto
13     retorna dadosDeAvaliacao;
14 fim
```

Na avaliação do modelo, foram comparados sete modelos de aprendizado de máquina com foco em regressão. Esses modelos foram abreviados por questões de espaço, sendo apresentados da seguinte forma: *Support Vector Regressor* com *kernels* RBF (SVR-R), Sigmoide (SVR-S) e linear (SVR-L), *K-Nearest Neighbors* (KNN), *Decision Tree* (DCT), Lasso e *Linear Regression* (LR). Com a finalidade de permitir uma comparação mais abrangente entre os modelos, também foi implementado o algoritmo proposto por

Liu, Pan e Liu (2019), que se baseia no monitoramento de objetos em um período de janela fixo de um mês e utiliza um limite de número de acessos como uma solicitação.

Neste trabalho foi proposto um modelo de aprendizado de máquina supervisionado com o objetivo de prever a quantidade de acessos futuros de objetos e utilizar um limite de acessos calculado para determinar a classe de cada objeto. A estrutura de preços do armazenamento em nuvem apresentada na Tabela 4.1 foi adotada, onde o limite de número de acessos é igual a um. Nesse sentido, a classificação de cada objeto é determinada comparando o número de acessos previsto com esse limite. No entanto, uma abordagem adicional foi empregada para variar esse limite em 50% a mais e a menos, com o intuito de comparar o desempenho do modelo em diferentes cenários e explorar novas abordagens de classificação.

Por fim, neste trabalho foram utilizadas três configurações diferentes de janelamento de tempo, com o objetivo de compreender como os padrões de acesso influenciam na previsão. A primeira configuração adotada foi um janelamento com período e tamanho do passo iguais a uma semana. A segunda configuração consistiu em um janelamento com período igual a quatro semanas e tamanho do passo igual a uma semana. Já a terceira configuração envolveu um janelamento com período e tamanho do passo iguais a quatro semanas. A variação dessas configurações permitiu analisar os efeitos dos diferentes intervalos de tempo na capacidade de previsão do modelo.

5 Experimentos e Resultados

Neste capítulo é apresentada uma análise abrangente dos resultados obtidos com o modelo proposto. São fornecidas informações sobre a origem e características da base de dados, bem como a forma como ela foi preparada para ser utilizada. Além disso, são apresentados os resultados obtidos, incluindo métricas de avaliação e interpretação dos resultados.

5.1 Bases de Dados

Foram utilizados neste estudo traços reais de acessos a dados armazenados no Dropbox, coletados por Gonçalves et al. (2016) a partir de dois pontos de presença (PoPs) de provedores de serviço de internet residenciais, denominados PoP-1 e PoP-2. Esses traços registram os acessos dos usuários aos seus dados por meio desses PoPs, sendo que cada dado é representado por um Dropbox *namespace*, que é uma estrutura utilizada no serviço para identificar de forma exclusiva um arquivo de usuário (documento, áudio, imagem) ou um diretório. Portanto, assumiu-se que cada *namespace* corresponde a um objeto que o usuário acessa no serviço de armazenamento em nuvem.

Os acessos dos usuários são representados por identificadores anônimos (ID), que não revelam informações sobre a identidade dos usuários ou o conteúdo armazenado no serviço, mas permitem a análise do padrão de acesso aos dados no Dropbox. As informações sobre os acessos dos usuários utilizadas incluem o registro de data/hora do acesso, o ID do objeto e uma estimativa do seu volume. Essa estimativa de volume é calculada somando-se os bytes transferidos (upload e download) para cada objeto, proporcionando uma estimativa conservadora para cada objeto no conjunto de dados.

Com base nesses dados, foram construídas séries temporais para cada objeto, onde as unidades de tempo representam o número de acessos por semana. Além disso, foi estimado o volume desses objetos em cada unidade de tempo. Caso um objeto apresente volume zero em um determinado tempo, isso indica que o arquivo ainda não existia naquele momento. Ao todo, foram obtidas 32 semanas de dados para 12924 objetos na PoP-1 e

48 semanas de dados para 4987 objetos na PoP-2.

5.2 Métricas

As 11 métricas utilizadas para avaliação do desempenho dos modelos de aprendizado de máquinas são apresentadas no Capítulo 2. Vale destacar que, no cálculo da métrica F-beta, foi adotado um valor de β igual a 0,5, o que resulta em uma maior importância atribuída à precisão em relação à revocação.

Entre essas métricas, destaca-se a métrica *rcs* (economia de custo relativa), que desempenha um papel fundamental no trabalho, uma vez que o principal objetivo é alcançar a economia de custos para os usuários finais de serviços de armazenamento em nuvem, sem comprometer a qualidade do serviço. Enquanto as outras métricas consideram apenas a previsão do número de acessos, o *rcs* leva em consideração também o volume dos objetos, o que tem um impacto significativo no cálculo dos custos de armazenamento.

A métrica *rcs* será avaliada com base em duas políticas de referência distintas. A primeira política, denominada rcs_{online}, é definida na Equação (5.1), onde N é o número total de períodos avaliados, e compara o custo do modelo proposto com o custo do algoritmo online proposto por Liu, Pan e Liu (2019). Essa política serve como um ponto de referência para avaliar o desempenho do modelo em relação a um método estabelecido na literatura.

$$rcs_{online} = \sum_{n=1}^{N} \frac{custo\_online_n - custo\_do\_modelo_n}{custo\_online_n} \tag{5.1}$$

A segunda política, denominada rcs_{allHot}, é indicada na Equação (5.2). Nessa política, o custo do modelo proposto é comparado com uma política de não migração entre classes, na qual todos os objetos são mantidos na classe mais frequente. Essa política representa uma abordagem conservadora que evita a migração de objetos entre classes, e seu custo serve como uma referência adicional para a avaliação do modelo.

$$rcs_{allHot} = \sum_{n=1}^{N} \frac{custo\_all\_hot_n - custo\_do\_modelo_n}{custo\_all\_hot_n} \tag{5.2}$$

Ao utilizar essas duas políticas de referência, é possível verificar se o modelo é ca-

paz de proporcionar uma economia de custos significativa, enquanto mantém a qualidade do serviço de armazenamento em nuvem.

5.3 Resultados

Nesta seção são apresentados os resultados do estudo que analisou os efeitos da aplicação do modelo proposto nas bases de dados PoP-1 e PoP-2. Os resultados são organizados em seis tabelas distintas, correspondendo a cada uma das bases de dados consideradas. Cada tabela contém informações sobre as métricas de desempenho e os custos de armazenamento final do modelo, sendo três tabelas para as métricas e três tabelas para os custos.

As três tabelas de métricas apresentam os resultados das métricas de desempenho apresentadas no Capítulo 2. Cada tabela corresponde a um dos janelamentos de tempo utilizados no estudo, ou seja, 1×1, 4×1 e 4×4. Para cada algoritmo utilizado, os resultados são apresentados para os cinco limites de acessos determinados, permitindo uma análise comparativa do desempenho em uma variação do valor obtido do modelo de custos. É importante destacar que, para o cálculo das métricas de classificação, foi considerada a classe "frequente" como classe positiva e a classe "infrequente" como classe negativa.

As três tabelas de custos fornecem informações sobre os custos totais de armazenamento dos objetos das bases de dados, além das métricas de custo *rcs*. Esses custos levam em consideração os preços estabelecidos na Tabela 4.1 e as classes atribuídas pelo modelo proposto. Além disso, são apresentados o custo total ótimo de armazenamento, que corresponde à situação em que o modelo acerta a melhor classe em cada decisão, o custo total de manter todos os objetos na classe quente, denominado (*always hot*), e o custo total da aplicação do algoritmo online proposto por Liu, Pan e Liu (2019). Assim como nas tabelas de métricas, as tabelas de custos são segmentadas de acordo com os janelamentos de tempo utilizados, apresentando os resultados para cada algoritmo e para os cinco limites de acessos testados.

As Tabelas 5.1, 5.2 e 5.3 apresentam os resultados das métricas para a base PoP-1, enquanto as Tabelas 5.7, 5.8 e 5.9 apresentam os custos. Já as Tabelas 5.4, 5.5 e 5.6 indicam os resultados das métricas para a base PoP-2 e as Tabelas 5.10, 5.11 e 5.12

indicam os custos.

Tabela 5.1: Resultados das métricas nos teste na base PoP-1 com o janelamento 1 × 1.

Algoritmo	Limite	MAE	MSE	RMSE	R^2	Acurácia	Precisão	Revocação	F_1	$F_{0.5}$	AUC
DCT	0.5	8.79	8084.01	89.91	-0.844	0.30	0.29	**0.96**	0.43	0.33	0.50
	0.75	8.79	8084.01	89.91	-0.844	0.32	0.31	0.95	0.44	0.34	0.51
	1	8.79	8084.01	89.91	-0.844	0.35	0.32	0.93	0.44	0.36	0.52
	1.25	8.79	8084.01	89.91	-0.844	0.42	0.37	0.88	0.47	0.40	0.55
	1.5	8.79	8084.01	89.91	-0.844	**0.57**	**0.46**	0.77	**0.53**	**0.48**	**0.62**
KNN	0.5	12.20	5519.32	74.29	-0.299	0.67	0.54	**0.67**	**0.55**	0.54	**0.66**
	0.75	12.20	5519.32	74.29	-0.299	**0.67**	0.56	0.64	0.54	**0.54**	0.66
	1	12.20	5519.32	74.29	-0.299	0.67	0.56	0.62	0.52	0.53	0.65
	1.25	12.20	5519.32	74.29	-0.299	0.67	0.57	0.61	0.52	0.53	0.64
	1.5	12.20	5519.32	74.29	-0.299	0.67	**0.57**	0.61	0.52	0.53	0.64
LR	0.5	8.45	6731.78	82.05	-0.165	0.30	0.29	**0.98**	0.43	0.33	0.51
	0.75	8.45	6731.78	82.05	-0.165	0.34	0.33	0.94	0.44	0.36	0.52
	1	8.45	6731.78	82.05	-0.165	0.36	0.34	0.92	0.44	0.36	0.53
	1.25	8.45	6731.78	82.05	-0.165	0.39	0.35	0.90	0.45	0.37	0.54
	1.5	8.45	6731.78	82.05	-0.165	**0.42**	**0.37**	0.88	**0.46**	**0.38**	**0.55**
LASSO	0.5	8.83	6843.58	82.73	-0.168	0.30	0.29	**0.98**	0.43	0.33	0.51
	0.75	8.83	6843.58	82.73	-0.168	0.32	0.31	0.95	0.44	0.34	0.51
	1	8.83	6843.58	82.73	-0.168	0.34	0.33	0.93	0.44	0.35	0.52
	1.25	8.83	6843.58	82.73	-0.168	0.37	0.34	0.92	0.45	0.36	0.53
	1.5	8.83	6843.58	82.73	-0.168	**0.42**	**0.37**	0.88	**0.46**	**0.38**	**0.55**
SVR-L	0.5	6.03	4695.24	68.52	0.224	0.79	0.66	**0.52**	**0.56**	0.60	**0.70**
	0.75	6.03	4695.24	68.52	0.224	**0.80**	0.71	0.44	0.52	**0.60**	0.68
	1	6.03	4695.24	68.52	0.224	0.79	0.72	0.40	0.50	0.60	0.67
	1.25	6.03	4695.24	68.52	0.224	0.79	0.74	0.37	0.48	0.59	0.66
	1.5	6.03	4695.24	68.52	0.224	0.79	**0.75**	0.34	0.45	0.58	0.65
SVR-R	0.5	6.27	6043.70	77.74	0.026	**0.80**	0.71	**0.45**	**0.53**	**0.61**	**0.69**
	0.75	6.27	6043.70	77.74	0.026	0.80	0.75	0.38	0.49	0.60	0.66
	1	6.27	6043.70	77.74	0.026	0.79	0.76	0.34	0.46	0.59	0.65
	1.25	6.27	6043.70	77.74	0.026	0.79	0.77	0.31	0.43	0.57	0.64
	1.5	6.27	6043.70	77.74	0.026	0.78	**0.78**	0.28	0.41	0.56	0.63
SVR-S	0.5	11.80	10786.10	103.86	-1.472	0.77	0.58	**0.56**	**0.56**	0.57	**0.70**
	0.75	11.80	10786.10	103.86	-1.472	0.77	0.58	0.56	0.56	0.57	0.70
	1	11.80	10786.10	103.86	-1.472	0.77	0.58	0.56	0.56	0.57	0.70
	1.25	11.80	10786.10	103.86	-1.472	0.77	0.59	0.55	0.56	0.57	0.70
	1.5	11.80	10786.10	103.86	-1.472	**0.77**	**0.59**	0.55	0.56	**0.58**	0.70

É relevante salientar que as métricas de regressão geralmente não são afetadas pela variação do limite de acessos, com exceção de certos casos em que o algoritmo *Decision Tree* toma decisões internas que podem levar a alterações nessas métricas. Essa falta de variação ocorre devido ao fato de que o limite de acessos é utilizado apenas na fase de binarização do modelo, ou seja, na etapa em que os acessos futuros previstos são usados para classificar cada objeto.

Pelos resultados apresentados nas tabelas de métricas, é evidente que, independentemente do modelo de regressão utilizado, observamos médias de erros elevadas e valores baixos para o coeficiente de determinação (R^2 score), em alguns casos até valores

Tabela 5.2: Resultados das métricas nos teste na base PoP-1 com o janelamento 4 × 1.

Algoritmo	Limite	MAE	MSE	RMSE	R^2	Acurácia	Precisão	Revocação	F_1	$F_{0.5}$	AUC
DCT	0.5	28.07	89086.81	298.47	-33.876	0.29	**0.27**	**0.92**	**0.41**	**0.31**	**0.48**
	0.75	27.54	69981.95	264.54	-30.279	0.29	0.27	0.92	0.41	0.31	0.48
	1	27.63	68629.07	261.97	-33.123	0.29	0.27	0.89	0.40	0.31	0.47
	1.25	28.37	91491.35	302.48	-32.402	0.29	0.27	0.88	0.40	0.31	0.47
	1.5	28.29	90529.77	300.88	-35.141	**0.30**	0.27	0.88	0.40	0.31	0.47
KNN	0.5	20.89	22462.24	149.87	-10.339	0.59	0.41	**0.84**	0.54	0.45	0.66
	0.75	20.89	22462.24	149.87	-10.339	0.59	0.41	0.83	0.54	0.45	0.66
	1	20.89	22462.24	149.87	-10.339	0.63	0.43	0.80	0.55	0.47	0.68
	1.25	20.89	22462.24	149.87	-10.339	0.63	0.44	0.79	**0.55**	0.48	0.68
	1.5	20.89	22462.24	149.87	-10.339	**0.64**	**0.44**	0.78	0.55	**0.48**	**0.68**
LR	0.5	27.21	76352.45	276.32	-27.714	0.30	0.28	**0.98**	0.43	0.33	0.51
	0.75	27.21	76352.45	276.32	-27.714	0.30	0.28	0.98	0.43	0.33	0.51
	1	27.21	76352.45	276.32	-27.714	0.32	0.29	0.97	0.44	0.34	0.52
	1.25	27.21	76352.45	276.32	-27.714	0.32	0.29	0.96	0.44	0.34	0.52
	1.5	27.21	76352.45	276.32	-27.714	**0.32**	**0.30**	0.96	**0.44**	**0.34**	**0.52**
LASSO	0.5	26.79	69378.70	263.40	-26.046	0.30	0.28	**0.99**	0.43	0.33	0.51
	0.75	26.79	69378.70	263.40	-26.046	0.30	0.28	0.99	0.43	0.33	0.51
	1	26.79	69378.70	263.40	-26.046	0.31	0.29	0.97	0.44	0.34	0.52
	1.25	26.79	69378.70	263.40	-26.046	0.32	0.29	0.97	**0.44**	0.34	0.52
	1.5	26.79	69378.70	263.40	-26.046	**0.32**	**0.30**	0.96	0.44	**0.34**	**0.52**
SVR-L	0.5	12.33	13724.71	117.15	-7.855	0.69	0.48	**0.77**	0.58	0.51	0.71
	0.75	12.33	13724.71	117.15	-7.855	0.71	0.50	0.75	0.58	0.53	0.71
	1	12.33	13724.71	117.15	-7.855	0.74	0.52	0.71	0.59	0.55	0.73
	1.25	12.33	13724.71	117.15	-7.855	0.75	0.54	0.68	0.59	0.56	**0.73**
	1.5	12.33	13724.71	117.15	-7.855	**0.76**	**0.56**	0.66	**0.59**	**0.57**	0.73
SVR-R	0.5	7.23	5924.53	76.97	0.039	0.66	0.48	**0.78**	0.57	0.51	0.69
	0.75	7.23	5924.53	76.97	0.039	0.68	0.51	0.74	0.58	0.53	0.69
	1	7.23	5924.53	76.97	0.039	0.72	0.54	0.69	**0.58**	0.55	0.70
	1.25	7.23	5924.53	76.97	0.039	0.73	0.56	0.66	0.58	0.56	0.70
	1.5	7.23	5924.53	76.97	0.039	**0.75**	**0.58**	0.63	0.58	**0.57**	**0.70**
SVR-S	0.5	8.56	6535.19	80.84	-0.137	0.63	0.42	**0.83**	0.55	0.47	0.68
	0.75	8.56	6535.19	80.84	-0.137	0.65	0.44	0.80	0.56	0.48	0.69
	1	8.56	6535.19	80.84	-0.137	0.68	0.46	0.78	0.57	0.50	0.70
	1.25	8.56	6535.19	80.84	-0.137	0.70	0.49	0.75	0.58	0.52	0.71
	1.5	8.56	6535.19	80.84	-0.137	**0.71**	**0.50**	0.73	**0.58**	**0.53**	**0.71**

negativos.

Esses resultados indicam que os modelos não conseguem prever de forma precisa a quantidade de acessos futuros, especialmente considerando que possuem apenas uma variável independente, ou seja, o número de acessos passado. No entanto, é interessante notar que alguns modelos de regressão apresentam métricas favoráveis relacionadas à classificação posterior realizada. Isso significa que, embora não sejam precisos na previsão exata dos acessos futuros, eles são capazes de classificar os objetos corretamente com base em um limite de referência, o que é crucial para o contexto do problema em questão.

É observado que os algoritmos *Decision Tree*, *Linear Regression* e Lasso apresentam valores de revocação altos, porém possuem baixa precisão, baixa acurácia e, con-

Tabela 5.3: Resultados das métricas nos teste na base PoP-1 com o janelamento 4 × 4.

Algoritmo	Limite	MAE	MSE	RMSE	R^2	Acurácia	Precisão	Revocação	F_1	$F_{0.5}$	AUC
DCT	0.5	35.94	102792.23	320.61	-1.093	0.45	0.45	**0.92**	**0.60**	**0.50**	0.48
	0.75	35.68	80056.78	282.94	-0.577	**0.45**	**0.45**	0.92	0.60	0.50	**0.48**
	1	37.03	129800.02	360.28	-1.625	0.44	0.45	0.90	0.59	0.50	0.47
	1.25	35.98	84681.83	291.00	-0.750	0.44	0.45	0.88	0.59	0.49	0.47
	1.5	36.06	92147.99	303.56	-0.825	0.44	0.45	0.88	0.59	0.49	0.47
KNN	0.5	28.25	52454.93	229.03	0.230	0.62	0.58	**0.86**	**0.68**	0.62	0.62
	0.75	28.25	52454.93	229.03	0.230	0.63	0.59	0.84	0.68	0.62	0.62
	1	28.25	52454.93	229.03	0.230	0.67	0.64	0.76	0.68	0.65	**0.66**
	1.25	28.25	52454.93	229.03	0.230	**0.67**	0.65	0.74	0.67	**0.66**	0.66
	1.5	28.25	52454.93	229.03	0.230	0.67	**0.65**	0.73	0.67	0.66	0.65
LR	0.5	32.09	106099.31	325.73	-0.853	0.46	0.46	**0.99**	**0.63**	0.52	0.50
	0.75	32.09	106099.31	325.73	-0.853	0.46	0.46	0.99	0.63	0.52	0.50
	1	32.09	106099.31	325.73	-0.853	0.46	0.46	0.99	0.63	0.52	0.50
	1.25	32.09	106099.31	325.73	-0.853	0.46	**0.46**	0.99	0.63	**0.52**	**0.50**
	1.5	32.09	106099.31	325.73	-0.853	**0.46**	0.46	0.99	0.63	0.52	0.50
LASSO	0.5	31.62	91863.79	303.09	-0.565	0.46	0.46	**1.00**	**0.63**	0.52	0.50
	0.75	31.62	91863.79	303.09	-0.565	**0.46**	**0.46**	1.00	0.63	**0.52**	**0.50**
	1	31.62	91863.79	303.09	-0.565	0.46	0.46	1.00	0.63	0.52	0.50
	1.25	31.62	91863.79	303.09	-0.565	0.46	0.46	1.00	0.63	0.52	0.50
	1.5	31.62	91863.79	303.09	-0.565	0.46	0.46	1.00	0.63	0.52	0.50
SVR-L	0.5	22.08	43033.06	207.44	0.420	**0.73**	0.73	**0.68**	**0.70**	0.71	**0.73**
	0.75	22.08	43033.06	207.44	0.420	0.73	0.74	0.64	0.68	0.72	0.72
	1	22.08	43033.06	207.44	0.420	0.73	0.76	0.62	0.68	**0.72**	0.72
	1.25	22.08	43033.06	207.44	0.420	0.73	0.77	0.59	0.66	0.72	0.71
	1.5	22.08	43033.06	207.44	0.420	0.73	**0.78**	0.57	0.65	0.72	0.71
SVR-R	0.5	23.69	61346.92	247.68	0.019	0.67	0.68	**0.75**	**0.68**	0.67	0.66
	0.75	23.69	61346.92	247.68	0.019	0.66	0.69	0.71	0.66	0.67	0.65
	1	23.69	61346.92	247.68	0.019	**0.70**	0.74	0.63	0.65	0.69	**0.67**
	1.25	23.69	61346.92	247.68	0.019	0.70	0.76	0.60	0.63	0.69	0.67
	1.5	23.69	61346.92	247.68	0.019	0.70	**0.77**	0.57	0.62	**0.69**	0.66
SVR-S	0.5	24.01	63568.31	252.13	0.004	0.70	0.65	**0.79**	**0.71**	0.67	0.69
	0.75	24.01	63568.31	252.13	0.004	0.71	0.67	0.76	0.70	0.68	0.69
	1	24.01	63568.31	252.13	0.004	0.71	0.69	0.73	0.70	0.69	**0.69**
	1.25	24.01	63568.31	252.13	0.004	0.71	0.71	0.69	0.68	0.70	0.69
	1.5	24.01	63568.31	252.13	0.004	**0.71**	**0.72**	0.68	0.68	**0.70**	0.68

sequentemente, valores baixos de AUC-ROC. Essa tendência é devido ao fato de que esses algoritmos têm uma propensão a classificar a maioria dos objetos como “frequente”, o que resulta em custos semelhantes à estratégia de manter todos os objetos na classe “frequente”.

Em contraste, os algoritmos *K-Nearest Neighbors* e *Support Vector Regressor* mostram valores menores de revocação, mas apresentam maior acurácia, precisão e, portanto, valores mais altos de AUC-ROC. Isso significa que, embora esses algoritmos possam não acertar tanto os objetos na classe “frequente”, eles cometem menos erros ao classificar os objetos na classe “infrequente”, resultando em maior economia de custos.

Observa-se que, em relação ao limite de acessos, as métricas de precisão, acurácia

Tabela 5.4: Resultados das métricas nos teste na base PoP-2 com o janelamento 1 × 1.

Algoritmo	Limite	MAE	MSE	RMSE	R^2	Acurácia	Precisão	Revocação	F_1	$F_{0.5}$	AUC
DCT	0.5	15.50	18449.26	135.83	-0.023	0.32	0.31	**0.97**	0.45	0.35	0.50
	0.75	15.50	18449.26	135.83	-0.023	0.37	0.34	0.94	0.47	0.38	0.52
	1	15.50	18449.26	135.83	-0.023	0.38	0.35	0.93	0.48	0.39	0.52
	1.25	15.50	18449.26	135.83	-0.023	0.46	0.40	0.87	0.51	0.43	0.56
	1.5	15.50	18449.26	135.83	-0.023	**0.57**	**0.48**	0.80	**0.55**	**0.50**	**0.60**
KNN	0.5	16.91	17053.87	130.59	0.227	0.57	0.50	**0.80**	0.56	0.51	0.62
	0.75	16.91	17053.87	130.59	0.227	0.59	0.51	0.79	0.56	0.52	0.63
	1	16.91	17053.87	130.59	0.227	0.60	0.52	0.76	0.56	0.53	0.63
	1.25	16.91	17053.87	130.59	0.227	0.61	0.53	0.75	0.56	0.54	0.63
	1.5	16.91	17053.87	130.59	0.227	**0.63**	**0.55**	0.73	**0.56**	**0.54**	**0.64**
LR	0.5	14.46	21926.99	148.08	-0.009	0.36	0.35	**0.95**	0.47	0.38	0.52
	0.75	14.46	21926.99	148.08	-0.009	0.41	0.38	0.92	0.49	0.41	0.54
	1	14.46	21926.99	148.08	-0.009	0.44	0.40	0.90	0.50	0.43	0.55
	1.25	14.46	21926.99	148.08	-0.009	0.45	0.42	0.89	0.51	0.44	0.56
	1.5	14.46	21926.99	148.08	-0.009	**0.47**	**0.43**	0.88	**0.51**	**0.45**	**0.56**
LASSO	0.5	15.27	16330.63	127.79	0.292	0.34	0.34	**0.97**	0.47	0.38	0.52
	0.75	15.27	16330.63	127.79	0.292	0.39	0.37	0.94	0.49	0.40	0.54
	1	15.27	16330.63	127.79	0.292	0.40	0.38	0.93	0.49	0.41	0.54
	1.25	15.27	16330.63	127.79	0.292	0.44	0.40	0.90	0.50	0.43	0.55
	1.5	15.27	16330.63	127.79	0.292	**0.46**	**0.42**	0.89	**0.51**	**0.44**	**0.56**
SVR-L	0.5	10.91	16578.97	128.76	0.301	0.79	0.68	**0.60**	**0.61**	0.63	**0.73**
	0.75	10.91	16578.97	128.76	0.301	0.80	0.72	0.54	0.59	0.64	0.72
	1	10.91	16578.97	128.76	0.301	**0.80**	0.76	0.49	0.57	**0.65**	0.71
	1.25	10.91	16578.97	128.76	0.301	0.80	0.77	0.47	0.56	0.64	0.70
	1.5	10.91	16578.97	128.76	0.301	0.80	**0.78**	0.45	0.55	0.64	0.69
SVR-R	0.5	12.02	20593.80	143.51	0.020	**0.80**	0.75	**0.51**	**0.59**	**0.66**	**0.71**
	0.75	12.02	20593.80	143.51	0.020	0.80	0.78	0.44	0.55	0.66	0.69
	1	12.02	20593.80	143.51	0.020	0.79	0.80	0.39	0.51	0.64	0.67
	1.25	12.02	20593.80	143.51	0.020	0.79	0.81	0.36	0.49	0.63	0.66
	1.5	12.02	20593.80	143.51	0.020	0.78	**0.82**	0.34	0.47	0.62	0.65
SVR-S	0.5	12.61	21181.48	145.54	-0.065	0.79	0.66	**0.60**	**0.61**	0.63	**0.73**
	0.75	12.61	21181.48	145.54	-0.065	0.79	0.66	0.59	0.61	0.63	0.73
	1	12.61	21181.48	145.54	-0.065	0.79	0.68	0.57	0.60	0.64	0.73
	1.25	12.61	21181.48	145.54	-0.065	0.79	0.70	0.54	0.59	0.64	0.72
	1.5	12.61	21181.48	145.54	-0.065	**0.79**	**0.70**	0.53	0.59	**0.64**	0.72

e valores de AUC-ROC tendem a ser maiores quando são utilizados limites maiores do que 1. Em alguns casos, as métricas mencionadas não variem significativamente de acordo com o limite de acessos. Isso pode indicar que o desempenho do modelo não é afetado de maneira significativa pela variação do valor escolhido para o limite de acessos, ou seja, as métricas permanecem relativamente estáveis em valores de limites próximos ao obtido pelo modelo de custo.

Observa-se que, em relação ao janelamento de tempo, as métricas avaliadas não apresentam uma variação significativa, com exceção da métrica de precisão. Nota-se que a precisão tende a ser maior no janelamento 4x4 em comparação ao janelamento 4x1.

Essa diferença pode ser atribuída à natureza dos dados e ao período de tempo

Tabela 5.5: Resultados das métricas nos teste na base PoP-2 com o janelamento 4 × 1.

Algoritmo	Limite	MAE	MSE	RMSE	R^2	Acurácia	Precisão	Revocação	F_1	$F_{0.5}$	AUC
DCT	0.5	49.40	171807.82	414.50	-16.814	0.31	0.29	**0.91**	0.43	0.34	0.48
	0.75	50.02	180029.88	424.30	-17.110	0.31	0.29	0.90	0.43	0.34	0.48
	1	48.84	161786.67	402.23	-15.071	0.34	0.31	0.85	**0.43**	0.35	0.49
	1.25	48.37	166933.51	408.57	-14.810	0.35	0.31	0.85	0.43	0.35	0.49
	1.5	48.53	155346.73	394.14	-12.466	**0.35**	**0.31**	0.84	0.43	**0.35**	**0.49**
KNN	0.5	35.40	43971.14	209.69	-3.998	0.53	0.41	**0.89**	0.54	0.46	0.62
	0.75	35.40	43971.14	209.69	-3.998	0.53	0.42	0.88	0.55	0.46	0.62
	1	35.40	43971.14	209.69	-3.998	0.57	0.44	0.85	0.56	0.48	0.64
	1.25	35.40	43971.14	209.69	-3.998	0.60	0.46	0.83	0.57	0.50	0.66
	1.5	35.40	43971.14	209.69	-3.998	**0.61**	**0.47**	0.82	**0.57**	**0.50**	**0.66**
LR	0.5	60.80	746332.18	863.91	-75.464	0.33	0.32	**0.97**	0.46	0.36	0.51
	0.75	60.80	746332.18	863.91	-75.464	0.34	0.32	0.97	0.47	0.36	0.52
	1	60.80	746332.18	863.91	-75.464	0.36	0.33	0.96	0.47	0.37	0.53
	1.25	60.80	746332.18	863.91	-75.464	0.36	0.33	0.95	0.47	0.37	0.53
	1.5	60.80	746332.18	863.91	-75.464	**0.37**	**0.33**	0.95	**0.47**	**0.37**	**0.53**
LASSO	0.5	57.45	507292.08	712.24	-42.153	0.33	0.32	**0.98**	0.46	0.36	0.51
	0.75	57.45	507292.08	712.24	-42.153	0.33	0.32	0.98	0.46	0.36	0.51
	1	57.45	507292.08	712.24	-42.153	0.35	0.33	0.97	0.47	0.37	0.52
	1.25	57.45	507292.08	712.24	-42.153	0.35	0.33	0.97	0.47	0.37	0.52
	1.5	57.45	507292.08	712.24	-42.153	**0.35**	**0.33**	0.97	**0.47**	**0.37**	**0.52**
SVR-L	0.5	33.15	119081.01	345.08	-10.769	0.71	0.52	**0.79**	0.61	0.55	0.73
	0.75	33.15	119081.01	345.08	-10.769	0.72	0.54	0.77	**0.62**	0.56	0.73
	1	33.15	119081.01	345.08	-10.769	0.73	0.55	0.75	0.62	0.58	0.74
	1.25	33.15	119081.01	345.08	-10.769	0.74	0.57	0.73	0.62	0.58	0.74
	1.5	33.15	119081.01	345.08	-10.769	**0.75**	**0.58**	0.71	**0.62**	**0.59**	**0.74**
SVR-R	0.5	12.52	20340.18	142.62	0.038	0.66	0.52	**0.81**	0.61	0.55	0.68
	0.75	12.52	20340.18	142.62	0.038	0.69	0.55	0.77	0.61	0.57	0.68
	1	12.52	20340.18	142.62	0.038	0.73	0.60	0.69	0.61	0.60	**0.70**
	1.25	12.52	20340.18	142.62	0.038	0.74	0.63	0.65	0.61	0.61	0.70
	1.5	12.52	20340.18	142.62	0.038	**0.75**	**0.65**	0.62	0.60	**0.62**	0.70
SVR-S	0.5	12.49	19651.47	140.18	0.091	0.66	0.49	**0.84**	0.60	0.52	0.69
	0.75	12.49	19651.47	140.18	0.091	0.69	0.52	0.80	0.61	0.55	0.70
	1	12.49	19651.47	140.18	0.091	0.72	0.56	0.75	0.62	0.58	0.72
	1.25	12.49	19651.47	140.18	0.091	0.73	0.58	0.72	**0.62**	0.59	**0.72**
	1.5	12.49	19651.47	140.18	0.091	**0.74**	**0.60**	0.70	0.62	**0.60**	0.72

considerado em cada janela. No janelamento 4x4, onde são consideradas quatro semanas consecutivas, a probabilidade de ocorrer mais de um acesso durante esse período é maior em comparação ao janelamento 4x1, que abrange apenas uma semana.

Em relação aos custos, é interessante observar que os algoritmos K-NN e SVR, especialmente com *kernels* RBF (SVR-R) e linear (SVR-L), se destacam ao oferecer os menores custos e, portanto, proporcionam as melhores economias para os usuários.

Ao analisar os custos nos diferentes janelamentos de tempo, nota-se que os modelos treinados com um período de quatro semanas apresentam custos menores em comparação aos treinados em apenas uma semana. Essa diferença sugere que o padrão temporal dos acessos dos usuários desempenha um papel importante na previsão dos custos.

Tabela 5.6: Resultados das métricas nos teste na base PoP-2 com o janelamento 4 × 4.

Algoritmo	Limite	MAE	MSE	RMSE	R^2	Acurácia	Precisão	Revocação	F_1	$F_{0.5}$	AUC
DCT	0.5	62.43	203986.05	451.65	-0.447	0.45	0.46	**0.91**	**0.60**	0.50	0.48
	0.75	61.86	206642.13	454.58	-0.447	0.45	0.46	0.90	0.60	0.51	0.48
	1	62.31	212806.38	461.31	-0.621	0.47	0.49	0.82	0.59	**0.52**	0.50
	1.25	62.27	201521.81	448.91	-0.126	0.47	**0.50**	0.81	0.58	0.52	0.50
	1.5	62.54	205390.82	453.20	-0.556	**0.47**	0.50	0.80	0.57	0.52	**0.50**
KNN	0.5	55.22	161850.66	402.31	0.229	0.62	0.61	**0.86**	**0.68**	0.63	0.61
	0.75	55.22	161850.66	402.31	0.229	0.62	0.61	0.85	0.68	0.63	0.60
	1	55.22	161850.66	402.31	0.229	0.64	0.63	0.81	0.68	0.64	0.62
	1.25	55.22	161850.66	402.31	0.229	0.66	0.66	0.77	0.68	0.66	0.64
	1.5	55.22	161850.66	402.31	0.229	**0.66**	**0.66**	0.77	0.68	**0.66**	**0.65**
LR	0.5	74.95	509563.30	713.84	-2.894	0.51	0.51	**0.95**	0.64	0.55	0.52
	0.75	74.95	509563.30	713.84	-2.894	0.51	0.51	0.95	0.64	0.55	0.52
	1	74.95	509563.30	713.84	-2.894	**0.56**	0.55	0.92	**0.66**	0.59	**0.55**
	1.25	74.95	509563.30	713.84	-2.894	0.56	0.55	0.92	0.66	0.59	0.55
	1.5	74.95	509563.30	713.84	-2.894	0.56	**0.55**	0.92	0.66	**0.59**	0.55
LASSO	0.5	74.05	496943.62	704.94	-2.819	0.51	0.51	**0.96**	0.64	0.55	0.52
	0.75	74.05	496943.62	704.94	-2.819	0.51	0.51	0.96	0.64	0.55	0.52
	1	74.05	496943.62	704.94	-2.819	0.56	0.55	0.92	**0.66**	0.59	**0.55**
	1.25	74.05	496943.62	704.94	-2.819	0.56	0.55	0.92	0.66	0.59	0.55
	1.5	74.05	496943.62	704.94	-2.819	**0.56**	**0.56**	0.92	0.66	**0.59**	0.55
SVR-L	0.5	48.86	168674.08	410.70	-0.102	0.72	0.73	**0.71**	**0.69**	0.71	0.71
	0.75	48.86	168674.08	410.70	-0.102	0.72	0.75	0.68	0.69	0.71	0.71
	1	48.86	168674.08	410.70	-0.102	0.72	0.75	0.67	0.68	0.71	0.71
	1.25	48.86	168674.08	410.70	-0.102	**0.72**	0.78	0.62	0.67	**0.72**	**0.72**
	1.5	48.86	168674.08	410.70	-0.102	0.72	**0.78**	0.60	0.66	0.72	0.72
SVR-R	0.5	49.02	210579.77	458.89	0.005	0.70	0.71	**0.76**	**0.70**	0.69	0.65
	0.75	49.02	210579.77	458.89	0.005	**0.72**	0.75	0.71	0.69	**0.71**	0.66
	1	49.02	210579.77	458.89	0.005	0.71	0.80	0.57	0.63	0.71	**0.68**
	1.25	49.02	210579.77	458.89	0.005	0.70	0.82	0.53	0.60	0.70	0.68
	1.5	49.02	210579.77	458.89	0.005	0.69	**0.83**	0.49	0.58	0.69	0.67
SVR-S	0.5	48.38	207735.09	455.78	0.027	0.71	0.71	**0.73**	**0.69**	0.70	0.69
	0.75	48.38	207735.09	455.78	0.027	0.70	0.74	0.68	0.67	0.70	0.69
	1	48.38	207735.09	455.78	0.027	**0.72**	0.77	0.63	0.66	**0.71**	**0.70**
	1.25	48.38	207735.09	455.78	0.027	0.71	0.79	0.60	0.65	0.71	0.70
	1.5	48.38	207735.09	455.78	0.027	0.71	**0.80**	0.58	0.64	0.71	0.69

O janelamento 4x1 tem uma ligeira vantagem em relação ao 4x4, devido à capacidade de ajustar a classe de um objeto a cada semana, o que pode resultar em ganhos de economia durante essas alterações.

No que diz respeito aos limites de acessos, é relevante destacar que, na base de dados PoP-1, os melhores custos são alcançados em limites de acessos próximos ou iguais a 1.5. Já na base PoP-2, os melhores custos são obtidos em limites de acessos próximos ou iguais a 1. Isso evidencia que a escolha do limite de acessos está fortemente relacionada às características específicas da base de dados utilizada. Cada base de dados pode ter um comportamento diferente em relação aos custos, e a seleção do limite adequado depende dessas características.

Tabela 5.7: Resultados dos custos nos teste na base PoP-1 com o janelamento 1 × 1.

Algoritmo	Limite	Custo total	rcs_{online}	rcs_{allHot}
Ótimo	*	1740.71	*	*
Online	*	2002.62	*	*
Always Hot	*	2198.24	*	*
DCT	0.5	2215.47	-10.63%	-0.78%
	0.75	2209.87	-10.35%	-0.53%
	1	2202.78	-9.99%	-0.21%
	1.25	2158.92	-7.80%	1.79%
	1.5	**2091.22**	**-4.42%**	**4.87%**
KNN	0.5	**2069.40**	**-3.33%**	**5.86%**
	0.75	2073.36	-3.53%	5.68%
	1	2090.89	-4.41%	4.88%
	1.25	2092.43	-4.48%	4.81%
	1.5	2094.58	-4.59%	4.72%
LR	0.5	2191.35	-9.42%	0.31%
	0.75	2180.82	-8.90%	0.79%
	1	2179.13	-8.81%	0.87%
	1.25	2168.01	-8.26%	1.38%
	1.5	**2157.10**	**-7.71%**	**1.87%**
LASSO	0.5	2191.35	-9.42%	0.31%
	0.75	2190.03	-9.36%	0.37%
	1	2183.53	-9.03%	0.67%
	1.25	2172.42	-8.48%	1.17%
	1.5	**2157.10**	**-7.71%**	**1.87%**
SVR-L	0.5	**2037.46**	**-1.74%**	**7.31%**
	0.75	2070.72	-3.40%	5.80%
	1	2091.57	-4.44%	4.85%
	1.25	2111.56	-5.44%	3.94%
	1.5	2126.16	-6.17%	3.28%
SVR-R	0.5	**2054.02**	**-2.57%**	**6.56%**
	0.75	2088.89	-4.31%	4.97%
	1	2120.80	-5.90%	3.52%
	1.25	2149.88	-7.35%	2.20%
	1.5	2172.91	-8.50%	1.15%
SVR-S	0.5	**2217.21**	**-10.72%**	**-0.86%**
	0.75	2217.21	-10.72%	-0.86%
	1	2217.54	-10.73%	-0.88%
	1.25	2218.59	-10.78%	-0.93%
	1.5	2220.07	-10.86%	-0.99%

Tabela 5.8: Resultados dos custos nos teste na base PoP-1 com o janelamento 4 × 1.

Algoritmo	Limite	Custo Total	rcs_{online}	rcs_{allhot}
Ótimo	*	1740.71	*	*
Online	*	2038.19	*	*
Always Hot	*	2198.24	*	*
DCT	0.5	2238.46	-9.83%	-1.83%
	0.75	**2235.79**	**-9.70%**	**-1.71%**
	1	2247.01	-10.25%	-2.22%
	1.25	2249.05	-10.35%	-2.31%
	1.5	2248.67	-10.33%	-2.29%
KNN	0.5	2060.20	-1.08%	6.28%
	0.75	2058.46	-0.99%	6.36%
	1	2048.30	-0.50%	6.82%
	1.25	2046.94	-0.43%	6.88%
	1.5	**2046.56**	**-0.41%**	**6.90%**
LR	0.5	2199.36	-7.91%	-0.05%
	0.75	2199.05	-7.89%	-0.04%
	1	2194.51	-7.67%	0.17%
	1.25	2194.43	-7.67%	0.17%
	1.5	**2193.54**	**-7.62%**	**0.21%**
LASSO	0.5	2198.67	-7.87%	-0.02%
	0.75	2198.57	-7.87%	-0.01%
	1	2193.96	-7.64%	0.19%
	1.25	2193.66	-7.63%	0.21%
	1.5	**2193.39**	**-7.61%**	**0.22%**
SVR-L	0.5	2025.10	0.64%	7.88%
	0.75	2019.00	0.94%	8.15%
	1	2011.66	1.30%	8.49%
	1.25	2008.46	1.46%	8.63%
	1.5	**2004.60**	**1.65%**	**8.81%**
SVR-R	0.5	2021.71	0.81%	8.03%
	0.75	2015.80	1.10%	8.30%
	1	2010.53	1.36%	8.54%
	1.25	2010.36	1.37%	8.55%
	1.5	**2008.39**	**1.46%**	**8.64%**
SVR-S	0.5	2113.98	-3.72%	3.83%
	0.75	2107.86	-3.42%	4.11%
	1	2099.70	-3.02%	4.48%
	1.25	2093.28	-2.70%	4.78%
	1.5	**2090.56**	**-2.57%**	**4.90%**

Tabela 5.9: Resultados dos custos nos teste na base PoP-1 com o janelamento 4 × 4.

Algoritmo	Limite	Custo Total	rcs_{online}	rcs_{allhot}
Ótimo	*	1847.95	*	*
Online	*	2035.52	*	*
Always Hot	*	2172.81	*	*
DCT	0.5	2212.32	-8.69%	-1.82%
	0.75	**2207.06**	**-8.43%**	**-1.58%**
	1	2224.16	-9.27%	-2.36%
	1.25	2224.28	-9.27%	-2.37%
	1.5	2218.01	-8.97%	-2.08%
KNN	0.5	2077.72	-2.07%	4.38%
	0.75	2073.19	-1.85%	4.58%
	1	2049.01	-0.66%	5.70%
	1.25	**2046.61**	**-0.54%**	**5.81%**
	1.5	2047.24	-0.58%	5.78%
LR	0.5	2181.39	-7.17%	-0.40%
	0.75	2181.37	-7.17%	-0.39%
	1	2181.24	-7.16%	-0.39%
	1.25	**2180.87**	**-7.14%**	**-0.37%**
	1.5	2181.21	-7.16%	-0.39%
LASSO	0.5	**2179.14**	**-7.06%**	**-0.29%**
	0.75	2179.17	-7.06%	-0.29%
	1	2179.23	-7.06%	-0.30%
	1.25	2179.23	-7.06%	-0.30%
	1.5	2179.25	-7.06%	-0.30%
SVR-L	0.5	2020.22	0.75%	7.02%
	0.75	2016.05	0.96%	7.21%
	1	2012.29	1.14%	7.39%
	1.25	2012.29	1.14%	7.39%
	1.5	**2010.13**	**1.25%**	**7.49%**
SVR-R	0.5	2030.26	0.26%	6.56%
	0.75	2028.17	0.36%	6.66%
	1	**2015.02**	**1.01%**	**7.26%**
	1.25	2015.95	0.96%	7.22%
	1.5	2017.29	0.90%	7.16%
SVR-S	0.5	2107.28	-3.53%	3.02%
	0.75	2101.52	-3.24%	3.28%
	1	2095.96	-2.97%	3.54%
	1.25	2090.43	-2.70%	3.79%
	1.5	**2088.48**	**-2.60%**	**3.88%**

Tabela 5.10: Resultados dos custos nos teste na base PoP-2 com o janelamento 1 × 1.

Algoritmo	Limite	Custo Total	rcs_{online}	rcs_{allhot}
Ótimo	*	1653.61	*	*
Online	*	1868.60	*	*
Always Hot	*	2015.92	*	*
DCT	0.5	2050.14	-9.72%	-1.70%
	0.75	2034.49	-8.88%	-0.92%
	1	2034.30	-8.87%	-0.91%
	1.25	2008.11	-7.47%	0.39%
	1.5	**1964.79**	**-5.15%**	**2.54%**
KNN	0.5	1933.44	-3.47%	4.09%
	0.75	**1925.54**	**-3.05%**	**4.48%**
	1	1934.76	-3.54%	4.03%
	1.25	1933.42	-3.47%	4.09%
	1.5	1932.65	-3.43%	4.13%
LR	0.5	1998.28	-6.94%	0.87%
	0.75	1985.18	-6.24%	1.52%
	1	1972.98	-5.59%	2.13%
	1.25	1966.52	-5.24%	2.45%
	1.5	**1961.79**	**-4.99%**	**2.69%**
LASSO	0.5	2004.51	-7.27%	0.57%
	0.75	1991.17	-6.56%	1.23%
	1	1985.79	-6.27%	1.49%
	1.25	1971.33	-5.50%	2.21%
	1.5	**1969.30**	**-5.39%**	**2.31%**
SVR-L	0.5	**1895.97**	**-1.46%**	**5.95%**
	0.75	1915.80	-2.53%	4.97%
	1	1933.32	-3.46%	4.10%
	1.25	1944.77	-4.08%	3.53%
	1.5	1958.20	-4.80%	2.86%
SVR-R	0.5	**1922.66**	**-2.89%**	**4.63%**
	0.75	1965.80	-5.20%	2.49%
	1	1991.40	-6.57%	1.22%
	1.25	2010.59	-7.60%	0.26%
	1.5	2027.87	-8.52%	-0.59%
SVR-S	0.5	**2037.58**	**-9.04%**	**-1.07%**
	0.75	2039.43	-9.14%	-1.17%
	1	2044.30	-9.40%	-1.41%
	1.25	2050.88	-9.75%	-1.73%
	1.5	2053.21	-9.88%	-1.85%

Tabela 5.11: Resultados dos custos nos teste na base PoP-2 com o janelamento 4 × 1.

Algoritmo	Limite	Custo Total	rcs_{online}	rcs_{allhot}
Ótimo	*	1653.61	*	*
Online	*	1875.70	*	*
Always Hot	*	2015.92	*	*
DCT	0.5	2052.51	-9.43%	-1.82%
	0.75	2054.62	-9.54%	-1.92%
	1	2055.65	-9.59%	-1.97%
	1.25	2054.06	-9.51%	-1.89%
	1.5	**2051.16**	**-9.35%**	**-1.75%**
KNN	0.5	1918.52	-2.28%	4.83%
	0.75	1915.85	-2.14%	4.96%
	1	1905.65	-1.60%	5.47%
	1.25	1897.93	-1.19%	5.85%
	1.5	**1896.72**	**-1.12%**	**5.91%**
LR	0.5	2029.23	-8.18%	-0.66%
	0.75	2025.01	-7.96%	-0.45%
	1	**2016.40**	**-7.50%**	**-0.02%**
	1.25	2016.52	-7.51%	-0.03%
	1.5	2017.34	-7.55%	-0.07%
LASSO	0.5	2014.36	-7.39%	0.08%
	0.75	2014.33	-7.39%	0.08%
	1	**2006.49**	**-6.97%**	**0.47%**
	1.25	2006.57	-6.98%	0.46%
	1.5	2006.57	-6.98%	0.46%
SVR-L	0.5	1867.48	0.44%	7.36%
	0.75	1866.10	0.51%	7.43%
	1	1863.51	0.65%	7.56%
	1.25	**1862.73**	**0.69%**	**7.60%**
	1.5	1863.36	0.66%	7.57%
SVR-R	0.5	1866.55	0.49%	7.41%
	0.75	**1862.40**	**0.71%**	**7.61%**
	1	1862.84	0.69%	7.59%
	1.25	1864.79	0.58%	7.50%
	1.5	1867.31	0.45%	7.37%
SVR-S	0.5	1880.94	-0.28%	6.70%
	0.75	1874.08	0.09%	7.04%
	1	1868.49	0.38%	7.31%
	1.25	**1864.02**	**0.62%**	**7.54%**
	1.5	1865.30	0.55%	7.47%

Tabela 5.12: Resultados dos custos nos teste na base PoP-2 com o janelamento 4 × 4.

Algoritmo	Limite	Custo Total	rcs_{online}	rcs_{allhot}
Ótimo	*	1742.12	*	*
Online	*	1867.62	*	*
Always Hot	*	1996.38	*	*
DCT	0.5	2036.48	-9.04%	-2.01%
	0.75	2055.82	-10.08%	-2.98%
	1	2046.75	-9.59%	-2.52%
	1.25	2038.90	-9.17%	-2.13%
	1.5	**2032.16**	**-8.81%**	**-1.79%**
KNN	0.5	1904.57	-1.98%	4.60%
	0.75	1903.99	-1.95%	4.63%
	1	1899.82	-1.72%	4.84%
	1.25	1894.53	-1.44%	5.10%
	1.5	**1893.81**	**-1.40%**	**5.14%**
LR	0.5	1983.25	-6.19%	0.66%
	0.75	**1983.04**	**-6.18%**	**0.67%**
	1	1949.39	-4.38%	2.35%
	1.25	1948.53	-4.33%	2.40%
	1.5	1948.49	-4.33%	2.40%
LASSO	0.5	1978.28	-5.93%	0.91%
	0.75	1978.30	-5.93%	0.91%
	1	**1943.81**	**-4.08%**	**2.63%**
	1.25	1944.19	-4.10%	2.61%
	1.5	1944.14	-4.10%	2.62%
SVR-L	0.5	1870.05	-0.13%	6.33%
	0.75	1869.42	-0.10%	6.36%
	1	**1869.64**	**-0.11%**	**6.35%**
	1.25	1871.81	-0.22%	6.24%
	1.5	1872.43	-0.26%	6.21%
SVR-R	0.5	1866.56	0.06%	6.50%
	0.75	**1863.86**	**0.20%**	**6.64%**
	1	1867.67	0.00%	6.45%
	1.25	1872.26	-0.25%	6.22%
	1.5	1880.09	-0.67%	5.82%
SVR-S	0.5	1867.53	0.00%	6.45%
	0.75	1862.88	0.25%	6.69%
	1	**1856.24**	**0.61%**	**7.02%**
	1.25	1857.12	0.56%	6.98%
	1.5	1859.45	0.44%	6.86%

6 Conclusão

Uma das principais vantagens do uso de modelos offline, como os adotados neste estudo, é a flexibilidade e a capacidade de processamento de um grande volume de dados. Ao treinar o modelo com os dados históricos de acesso, é possível analisar padrões e tendências que podem influenciar os futuros acessos, possibilitando uma melhor alocação de recursos e redução de custos. Além disso, o uso de algoritmos de aprendizado de máquina permite a exploração de relações complexas e não lineares entre as variáveis, o que pode levar a melhores previsões e resultados.

Em conclusão, os resultados obtidos revelam algumas tendências e considerações importantes no contexto da avaliação de modelos de aprendizado de máquina para previsão de acessos em serviços de armazenamento em nuvem.

Os modelos de regressão analisados neste estudo apresentaram dificuldades em prever com precisão a quantidade de acessos futuros, especialmente considerando que possuem apenas uma variável independente, que é o número de acessos passado. No entanto, é importante destacar que alguns modelos demonstraram métricas favoráveis relacionadas à classificação subsequente realizada. Essa capacidade de classificação dos modelos resultam em valores interessantes de custo total de armazenamento.

Em relação aos custos, os algoritmos SVR-R e SVR-L se destacaram ao proporcionar os menores custos e, consequentemente, as melhores economias para os usuários. Utilizando SVR-L na base PoP-1, foi alcançada uma economia de 1.65% em relação ao algoritmo Online e de 8.81% em relação a manter todos os objetos na classe “frequente”. Na base PoP-2, utilizando SVR-R, foi obtida uma economia de 0.71% em relação ao Online e de 7.61% em relação a manter todos os objetos na classe “frequente”.

Além disso, os modelos treinados com um período de quatro semanas apresentaram custos inferiores em comparação aos treinados em apenas uma semana, o que ressalta a importância do padrão temporal dos acessos dos usuários na previsão de custos. Nota-se que os modelos treinados no janelamento 4x1 proporcionaram uma melhor economia, devido à capacidade de ajustar a classe de um objeto a cada semana. Já a escolha do li-

mite de acessos está fortemente relacionada às características específicas da base de dados utilizada.

Esses resultados destacam a eficácia do modelo proposto na redução de custos e na obtenção de economias significativas para os usuários. Além disso, demonstram a importância de considerar o padrão temporal dos acessos e a seleção adequada do janelamento de tempo para melhorar o desempenho dos modelos de previsão de custos em serviços de armazenamento em nuvem.

6.1 Trabalhos Futuros

Trabalhos futuros incluem explorar diferentes funções de cálculo do limite de acessos, levando mais em consideração o volume do objeto. Outra possibilidade seria realizar experimentos com diferentes bases de dados para analisar o desempenho do modelo em cenários com características distintas. Diferentes bases de dados podem apresentar padrões de acesso variados, distribuição de tamanhos de objetos distintos e comportamentos de usuários diversos. Essa análise permitiria verificar a generalização do modelo e sua capacidade de lidar com diferentes contextos.

Por fim, uma direção promissora seria explorar a utilização de métodos de aprendizado profundo, como redes neurais, ao invés de métodos de aprendizado supervisionado tradicionais. Cada uma dessas direções de pesquisa oferece oportunidades significativas para melhorar o modelo proposto, expandir sua aplicabilidade e aprimorar a compreensão dos padrões de acesso em serviços de armazenamento.

Bibliografia

CISCO. Cisco global cloud index: Forecast and methodology. *2016-2021 White Paper*, 2019. Disponível em: ⟨https://www.cisco.com-DocumentID1513879861264127⟩.

ERRADI, A.; MANSOURI, Y. Online cost optimization algorithms for tiered cloud storage services. *Journal of Systems and Software*, v. 160, p. 110457, 2020. ISSN 0164-1212. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S0164121219302316⟩.

FACELI, K.; LORENA, A. C.; GAMA, J.; CARVALHO, A. C. P. d. L. F. d. *Inteligência artificial: uma abordagem de aprendizado de máquina*. [S.l.]: LTC, 2011.

GONçALVES, G.; DRAGO, I.; SILVA, A. P. C. da; VIEIRA, A. B.; ALMEIDA, J. M. The impact of content sharing on cloud storage bandwidth consumption. *IEEE Internet Computing*, v. 20, n. 4, p. 26–35, 2016.

HARRIS, C. R.; MILLMAN, K. J.; WALT, S. J. van der; GOMMERS, R.; VIRTANEN, P.; COURNAPEAU, D.; WIESER, E.; TAYLOR, J.; BERG, S.; SMITH, N. J.; KERN, R.; PICUS, M.; HOYER, S.; KERKWIJK, M. H. van; BRETT, M.; HALDANE, A.; RÍO, J. F. del; WIEBE, M.; PETERSON, P.; GÉRARD-MARCHANT, P.; SHEPPARD, K.; REDDY, T.; WECKESSER, W.; ABBASI, H.; GOHLKE, C.; OLIPHANT, T. E. Array programming with NumPy. *Nature*, Springer Science and Business Media LLC, v. 585, n. 7825, p. 357–362, set. 2020. Disponível em: ⟨https://doi.org/10.1038/s41586-020-2649-2⟩.

HSU, Y.-F.; IRIE, R.; MURATA, S.; MATSUOKA, M. A novel automated cloud storage tiering system through hot-cold data classification. *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, p. 492–499, 2018.

KUHKAN, M. A method to improve the accuracy of k-nearest neighbor algorithm. *International Journal of Computer Engineering and Information Technology*, Dorma Trading, Est. Publishing Manager, v. 8, n. 6, p. 90, 2016.

LIU, M.; PAN, L.; LIU, S. To transfer or not: An online cost optimization algorithm for using two-tier storage-as-a-service clouds. *IEEE Access*, IEEE, v. 7, p. 94263–94275, 2019.

LIU, M.; PAN, L.; LIU, S. Keep hot or go cold: A randomized online migration algorithm for cost optimization in staas clouds. *IEEE Transactions on Network and Service Management*, v. 18, n. 4, p. 4563–4575, 2021.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. In: *Sistemas Inteligentes Fundamentos e Aplicações*. 1. ed. Barueri-SP: Manole Ltda, 2003. p. 32. ISBN 85-204-168.

NORVIG; RUSSEL, P. Artificial intelligence: A modern approach. *Prentice Hall Upper Saddle River, NJ, USA: Rani, M., Nayak, R., & Vyas, OP (2015). An ontology-based adaptive personalized e-learning system, assisted by software agents on cloud storage. Knowledge-Based Systems*, v. 90, p. 33–48, 2002.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

RANSTAM, J.; COOK, J. Lasso regression. *Journal of British Surgery*, Oxford University Press, v. 105, n. 10, p. 1348–1348, 2018.

RIBEIRO, S.; GONçALVES, G.; SILVA, F.; VIEIRA, A.; ALMEIDA, J. Análise de um serviço virtual de armazenamento que explora classes de objetos na nuvem e padrões de acesso. In: *Anais do XIX Workshop em Desempenho de Sistemas Computacionais e de Comunicação*. Porto Alegre, RS, Brasil: SBC, 2020. p. 85–96. ISSN 2595-6167. Disponível em: ⟨https://sol.sbc.org.br/index.php/wperformance/article/view/11108⟩.

TEAM, T. pandas development. *pandas-dev/pandas: Pandas*. Zenodo, 2023. Disponível em: ⟨https://doi.org/10.5281/zenodo.7979740⟩.