Previsão da Classe de Frequência de Acesso de Objetos em Serviços de Armazenamento em Nuvem

Flávio A. A. Motta[1], Patrick R. P. Lemes[2], Glauber D. Golçalves[3], Heder S. Bernardino[2], Saulo M. Villela[2], Alex B. Vieira[1,2]

[1]Programa de Pós-Graduação em Ciência da Computação
Universidade Federal de Viçosa (UFV), Viçosa – MG, Brasil

[2]Departamento de Ciência da Computação
Universidade Federal de Juiz de Fora (UFJF), Juiz de Fora – MG, Brasil

[3]Campus Senador Helvído Nunes de Barros
Universidade Federal do Piauí (UFPI), Picos – PI, Brasil

flavio.motta@ufv.br, {patrick.rudgeri,heder}@ice.ufjf.br,

ggoncalves@ufpi.edu.br, {saulo.moraes,alex.borges}@ufjf.edu.br

Abstract. *Cloud storage services offer to domestic and corporate users advantages as backup, data replication in different locations, data sharing, and collaborative work. Additionally, providers of these services offer tiered cloud storage with multiple pricing options based on the level of storage used. In this work, we investigate a relevant aspect regarding costs of this service for users: predicting the data access class as frequent or infrequent and allocating it in a suitable storage tier. In this sense, we propose a machine learning framework that predicts the appropriate classes based on data access patterns. We evaluated the performance of this model through data trace-oriented of a real service. Data allocation methods used in the literature demonstrate an improvement potential of up to 41% over traditional cloud storage methods. Our results show that there is a potential for storage cost savings of up to 15.92% when compared to data allocation methods used in the literature.*

Resumo. *Os serviços de armazenamento em nuvem oferecem vantagens para usuários domésticos e corporativos como backup, replicação de dados em diferentes locais, compartilhamento de dados e trabalho colaborativo. Além disso, os provedores desses serviços oferecem armazenamento em nuvem hierárquico com múltiplas opções de preços baseado no nível de armazenamento utilizado. Neste artigo, investigamos um aspecto relevante sobre os custos desse serviço para os usuários: a previsão da classe de acesso ao dado como frequente ou infrequente e sua alocação em um nível de armazenamento adequado. Nesse sentido, propomos um modelo de aprendizado de máquina que prevê as classes apropriadas com base em padrões de acesso a dados. Avaliamos o desempenho desse modelo através de base orientadas a traços de dados de um serviço real. Métodos de alocação de dados utilizados na literatura demonstram um potencial de melhoria de até 41% em relação a métodos tradicionais de armazenamento em nuvem. Nossos resultados mostram que existe um potencial de economia no custo de armazenamento de até 15,92% quando comparado a métodos de alocação de dados utilizado na literatura.*

1. Introdução

Os serviços de armazenamento em nuvem são uma alternativa ao armazenamento de dados tradicional em dispositivos locais. Esses serviços oferecem *backup*, replicação de dados em diferentes locais, compartilhamento de dados e trabalho colaborativo. As vantagens oferecidas pelo armazenamento em nuvem para usuários domésticos e corporativos refletem as tendências de crescimento no uso desses serviços. Por exemplo, o mais recente índice global de computação em nuvem da Cisco prevê um volume de 1,3 ZB de dados armazenados na nuvem até 2022, representando um aumento de 4,6 vezes desde 2016 [Cisco 2019].

Os provedores de serviços de armazenamento oferecem armazenamento em nuvem em camadas com várias opções de preços com base em classes de frequência de acessos a objetos [Irie et al. 2018]. Um objeto, neste caso, é uma coleção lógica de bytes representando arquivos com métodos de acesso bem definidos e um foco especial em leituras e compartilhamento de dados [Mesnier et al. 2003]. Tipicamente, os usuários de serviços de armazenamento acessam com frequência objetos recentes. Os acessos a um objeto diminuem gradativamente ao longo do tempo, tornando-se infrequentes.

Assim, normalmente os provedores de armazenamento adotam camadas de armazenamento para objetos frequentes com preço maior por volume, mas de acesso rápido. Por outro lado, há camadas para objetos infrequentes, cujo preço por volume é menor mas com um custo adicional por acesso ao objeto. Há ainda camadas para objetos inativos, por exemplo, *backup*, com o menor preço por volume, mas uma alta latência de acesso.

Apesar do comportamento típico descrito acima, há objetos ativos que transitam entre acessos frequentes e infrequentes. Usuários do serviço em nuvem devem alocar seus objetos nas classes apropriadas, de acordo com seus padrões de acesso recentes, o mais rápido possível para reduzir custos e ter um serviço adequado. Ao manter um objeto infrequente classificado como frequente, perde-se a chance de economizar em armazenamento, visto que o custo de armazenamento por volume é maior na camada frequente. No entanto, alterar a classe de objetos de maneira errada pode aumentar os custos. Por exemplo, classificar um objeto acessado com frequência em uma camada de armazenamento infrequente aumenta os custos gerais de armazenamento, devido o custo adicional por acesso. A questão chave é então identificar, antecipadamente, em qual camada alocar um determinado objeto de usuário.

Otimizações para serviços de armazenamento em nuvem baseados em acesso a dados têm atraído pesquisas da indústria e da academia. Os primeiros esforços de pesquisa se concentraram na otimização da infraestrutura de nuvem dos provedores [Kaushik and Bhandarkar 2010, Subramanian et al. 2014, Irie et al. 2018]. Dado o desenvolvimento dessa infraestrutura em direção ao armazenamento em camadas de acordo com a frequência de acesso a objetos, as pesquisas mais recentes desenvolveram métodos para que os usuários escolham adequadamente a classe de seus objetos. Esses métodos focam em algoritmos *online* [Liu et al. 2019, Liu et al. 2021, Erradi and Mansouri 2020], que rastreiam os acessos a cada objeto individualmente para definir sua camada, sem explorar a previsão de classe baseado nos padrões de acesso de vários objetos conjuntamente.

Neste artigo, propomos um modelo preditivo para que usuários possam prever a classe de seus objetos ainda ativos e os alocar adequadamente nas camadas frequen-

tes ou infrequentes de provedores de armazenamento em nuvem. Nossa proposta conta com técnicas de aprendizado de máquina para explorar os padrões de acesso aos objetos. Realizamos simulações orientadas a traços com base nos acessos de usuários coletados de um serviço de armazenamento real para avaliar nossa proposta. Especificamente, investigamos o limite de número de acessos frequentes extraídos da estrutura de preços e exploramos técnicas de aprendizado de máquina para prever classes de objetos no futuro próximo com base nesse limite. Nosso desafio de pesquisa é desenvolver uma estrutura para previsões rápidas usando os acessos mais recentes de objetos. Assim, enquanto os objetos estão ativos, os usuários podem aproveitar acesso de baixa latência oferecido pela maioria dos provedores em camadas de armazenamento frequentes e infrequentes e economizar custos ao mesmo tempo.

Os resultados apontam para oportunidades e desafios para aumentar os benefícios econômicos para os usuários dos serviços de armazenamento em nuvem. Em primeiro lugar, observamos analiticamente, por meio de nossos modelos de custos, que o modelo de precificação adotado pelos principais provedores nos permite calcular um limite no número de acessos em um período fixo, por exemplo, um mês, como meta de previsão dos benefícios econômicos para modificar a classe de objeto. Em seguida, avaliamos via experimentos um conjunto de classificadores para prever, de forma binária, se os acessos de um objeto excederão ou não o limite de acessos obtido. Nossas avaliações mostram que o modelo proposto alcança previsões com precisão acima de 75% e a economia nos custos de armazenamento em relação ao método encontrado na literatura [Liu et al. 2019, Erradi and Mansouri 2020] chega a 15,92%. Em suma, oferecemos duas importantes contribuições neste trabalho: (i) um modelo para previsão de classes de objetos, explorando seus padrões de acesso, visando otimizar os custos do serviço de armazenamento em nuvem para usuários, e (ii) um modelo de análise de custos de objetos, configurável de acordo com os preços praticados pelos provedores de armazenamento em nuvem.

2. Trabalhos Relacionados

A otimização de sistemas de armazenamento em nuvem com base no acesso aos dados tem atraído pesquisadores da indústria e da academia preocupados com o desempenho e os custos da infraestrutura de computadores.

Os primeiros esforços de pesquisa nesta área foram no desenvolvimento de infraestruturas de armazenamento em nuvem otimizadas para diferentes tipos de acessos de usuários. Neste tópico, pesquisadores do Yahoo [Kaushik and Bhandarkar 2010] analisaram três meses de traços de usuários na versão comercial do sistema Hadoop e propuseram a variação GreenHDFS. Essa variação de armazenamento aloca arquivos pouco frequentes para servidores em modo de baixo consumo de energia, reduzindo os custos de energia em 26%. Em [Subramanian et al. 2014], os autores projetaram uma camada de armazenamento intermediária na infraestrutura do Facebook, denominada f4, para lidar com objetos infrequentes, caracterizados como conteúdos gerados por usuários que não recebem mais acessos recentes. No entanto, os objetos infrequentes ainda estão ativos e recebem poucos acessos dos usuários. Em [Irie et al. 2018, Hsu et al. 2018], os autores propuseram a análise do acesso a diversos objetos por meio de aprendizado de máquina para dimensionar o tamanho das camadas de armazenamento em *data centers* em nuvem. Os autores aplicaram essa análise para classificar os arquivos de acordo com a previsão

de sua frequência de acesso. Em seguida, eles analisaram diferentes tamanhos de armazenamento para alocar os objetos, considerando a compensação entre desempenho e custo.

Todos os trabalhos acima se concentraram na otimização das infraestruturas de armazenamento em nuvem para o provedor de serviços. Ao contrário deles, focamos no lado do usuário e nos métodos para apoiar suas decisões. O desenvolvimento da infraestrutura de armazenamento em camadas direcionou as pesquisas mais recentes para desenvolver métodos para que os usuários escolham a classe de seus objetos corretamente.

Em [Liu et al. 2019, Erradi and Mansouri 2020, Liu et al. 2021], os autores propuseram algoritmos online que utilizam o número de acessos a um objeto dentro de períodos estáticos ou dinâmicos e retornam se devem mudar sua camada. A decisão é baseada em limites calculados a partir dos preços por volume e acessos de cada camada de armazenamento. Assim, um limite dado pelo número de acessos por período [Liu et al. 2019, Liu et al. 2021] ou um limite dado por período sem acesso [Erradi and Mansouri 2020] indica se deve mudar a camada do objeto.

Os autores comprovaram que ambas as estratégias atingem um ganho de custo mínimo garantido comparado a manter os objetos sempre em uma única camada. Estendemos essas estratégias aprendendo padrões de acesso para prever as classes apropriadas de objetos para cada camada de armazenamento, usando o mesmo período de curto prazo. Assim, mostramos que o modelo proposto neste trabalho pode superar a economia de custos dos algoritmos online.

A pesquisa sobre armazenamento em nuvem pessoal fornece importante contribuição sobre o comportamento dos usuários no serviço de armazenamento em nuvem e seus padrões de acesso. Em diversos estudos, os desempenhos de serviços como Dropbox, Onedrive e Google Drive foram analisados a partir da caracterização do comportamento dos usuários [Bocchi et al. 2015, Gracia-Tinedo et al. 2016, Gonçalves et al. 2016]. Esses trabalhos mostram importantes otimizações para serviços de armazenamento, mais especificamente com foco na redução de custos com transferências de dados e sincronização entre a nuvem e os dispositivos do usuário em compartilhamentos. Contudo, esses trabalhos não focam na análise de padrões de acesso para reduzir o custo dos usuários em serviços de armazenamento em nuvem.

3. Modelo

Nesta seção, descrevemos o modelo de previsão da classe de frequência de acesso de objetos em serviços de armazenamento em nuvem. Primeiro, apresentamos o modelo de custo de armazenamento considerado aqui. Este modelo de custo de armazenamento permite estimar os parâmetros para treinar o modelo de previsão de acordo com os preços dos serviços de armazenamento. Em seguida, mostramos o modelo de previsão que classifica os objetos com base nos padrões de acesso dos usuários.

3.1. Modelo de Custo

Os usuários acessam objetos recentes com frequência e, com o tempo, o número de acessos a esses objetos tende a diminuir gradativamente, ou seja, os acessos se tornam infrequentes [Subramanian et al. 2014]. Diante desse comportamento típico, usualmente, os provedores de serviços adotam uma estrutura de preços que varia de acordo com o padrão

Tabela 1. Principais componentes da estrutura de preços adotada pela maioria dos provedores de armazenamento em nuvem hierárquico: exemplo de Amazon S3 para três camadas com base nas classes dos objetos (frequente, infrequente e inativo). Preços de referência novembro 2021.

Componente	Frequente	Infrequente	Inativo
Volume (GB)	0.0230	0.0125	0.004
Operação (1K acessos)	0.0004	0.0010	0.050
Requisições (GB)	0.0000	0.0100	0.100

de acesso aos objetos. Assim, eles oferecem a camada de armazenamento padrão para objetos acessados com frequência e uma camada para objetos acessados com menor preço por volume, incluindo uma penalização de custo por acesso adicional. Essas camadas contrastam com as camadas de objetos inativos que geralmente têm alta latência (horas ou dias para acessar objetos) apesar do menor preços de acesso.

Tabela 1 apresenta os três componentes mais comuns do custo de armazenamento na Amazon, para os níveis de acesso frequente, infrequente e inativo. A Amazon é um dos maiores provedores de infraestrutura em nuvem atualmente com relação a seu principal serviço de armazenamento, chamado Simple Storage Service (S3).[1] Os componentes de custo de armazenamento são volume (em *bytes*), operações (*número de acessos*) e requisições (*bytes por acesso*). Outros grandes provedores de infraestrutura em nuvem, como Google e Microsoft, não mostrados na Tabela 1, seguem uma estrutura de preços semelhante.[2] De fato, a maioria dos provedores de infraestrutura segue essa estrutura com flutuações de preço de componentes aproximadamente proporcionais entre os níveis mostrados. Neste trabalho, focamos em classes de objetos frequentes e infrequentes para reduzir os custos sem o risco de degradação da qualidade do serviço aos usuários. Reiteramos que os principais provedores mantêm a melhor qualidade de serviço para essas classes, especificamente a menor latência para acessar objetos.

Baseado nessa estrutura de preço descrita, usamos o modelo de Ribeiro et al. 2020 para representar o custo para armazenar um objeto i na classe j na nuvem por período, definido como

$$C_{ij}(x_i, y_i) = v_j \times x_i + o_j \times y_i + r_j \times x_i \times y_i, \tag{1}$$

onde x_i é o volume e y_i é o número de acessos que o objeto i recebe por período. O custo para armazenar um objeto i na classe j é dado pelo seu volume v_j, custo de operação o_j (ou seja, o custo por acesso) e volume de requisição r_j (ou seja, o volume total de acesso em GB). Este modelo considera o armazenamento por período, que é omitido na formulação para fins de clareza. Novamente, por uma questão de simplificação, assumimos que as operações de leitura e gravação são igualmente importantes, e o modelo pode ser facilmente estendido para representar custos diferentes para elas.

Usamos a Equação (1) para estimar o limite do número de acessos que equivale ao preço para o mesmo objeto[3] armazenado em duas classes diferentes, seguindo uma

[1] https://aws.amazon.com/s3/pricing

[2] https://cloud.google.com/storage/pricing, https://azure.microsoft.com/pricing/details/storage/blobs

[3] Modificações em um objeto geram um novo objeto e os usuários de armazenamento em nuvem podem ter a opção de manter o histórico de versões de um objeto.

abordagem bastante semelhante àquela adotada em [Liu et al. 2019]. Seja $\hat{y}_i$ o limite de acessos, então igualamos os custos para duas classes distintas a e b por $C_{ia}(x_i, \hat{y}_i) = C_{ib}(x_i, \hat{y}_i)$ e obtemos o limite de acessos dado por:

$$\hat{y}_i = \frac{x_i \times (v_a - v_b)}{(o_b - o_a) + x_i \times (r_b - r_a)}, \tag{2}$$

onde a representa a classe onde os objetos são mais frequentes que b para um volume de objeto fixo x_i. Os parâmetros volume (v), operação (o) e requisição (r) são definidos de acordo com os custos oferecidos pelo provedor de infraestrutura.

A Figura 1 exemplifica o limite do número de acessos para um objeto de volume 1 GByte com base nos preços mostrados na Tabela 1. Neste exemplo, obtemos o limite $\hat{y}_i = 1$ acesso. Quando o número de acessos de um objeto for, em média, menor que 1 por um intervalo de tempo, ele deve ser classificado como infrequente. Caso contrário, a classe frequente (camada padrão) é a opção de baixo custo.

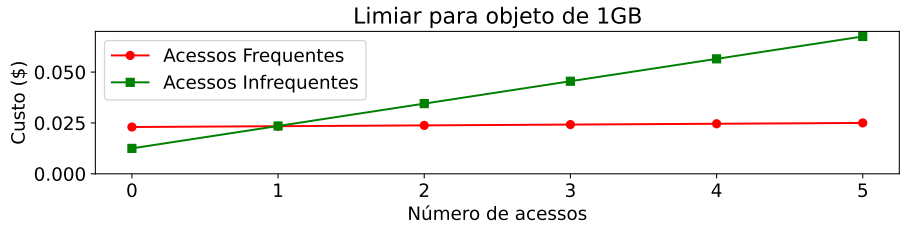

Figura 1. Limite de número de acesso $\hat{y}_i = 1$ para as classes frequentes (linha vermelha) e infrequentes (linha verde) considerando os preços mostrados na Tabela 1.

Com relação ao volume do objeto, podemos obter duas percepções interessantes da Equação (2). Primeiro, ao aumentar o volume dos objetos, tendemos a atingir um limite dado pela razão entre os componentes que representam custos por volume e requisição $\frac{v_a - v_b}{r_b - r_a}$. Observe que com os custos mostrados na Tabela 1 o valor do limite de acessos para utilizar a classe frequente é de um, independente do tamanho de um objeto. No entanto, um alto custo por volume para a classe a (v_a), ou seja, a classe frequente, pode aumentar esse limite para além do valor um. Em segundo lugar, tendemos a zerar o limite de acessos diminuindo o volume dos objetos, o que indica que a classe frequente é sempre a opção de menor custo para objetos muito pequenos. Na prática, os provedores de armazenamento em nuvem estabelecem um tamanho mínimo de objeto faturável. Tomando como exemplo o Amazon S3 na Tabela 1, esse tamanho é de 128 KBytes, o que também leva o limite para um valor próximo de um. Em suma, o modelo de custo fornece o limite do número de acessos que permite rotular as classes de objetos com base nos preços adotados pelo provedor de infraestrutura em nuvem e, em seguida, treinar o modelo de previsão.

3.2. Modelo de Previsão

O Algoritmo 1 apresenta uma visão geral da estrutura para prever classes de objetos, explorando padrões de acesso em um serviço de armazenamento na nuvem. A função de previsão utiliza dados de acesso para encontrar padrões, a fim de otimizar a operação do serviço. Esta função tem o papel mais importante do modelo em termos de custos e viabilidade do serviço de armazenamento, que é a previsão de acessos a objetos. Para realizar

as previsões, o componente pode usar diferentes métodos estatísticos, ou algoritmos de inteligência artificial, como por exemplo um comitê.

Algoritmo 1 Pseudocódigo do método proposto

Parâmetros: $arquivoDeAcessos$

1: $dadosTreino \leftarrow$ recuperaDadosTreino($arquivoDeAcessos$) {acesso de um período}
2: $dadosTeste \leftarrow$ recuperaDadosTeste($arquivoDeAcessos$) {acesso do próximo período}
3: **while** $dadosTeste \leq arquivoDeAcessos.tamanho$ **do**
4: $modelo =$ treinaModelo($dadosTreino$)
5: $custoTotal+ =$ testaModelo($dadosTreino$)
6: deslizaJanela($dadosTreino, dadosTeste$)
7: **fim while**
8: evaluateResult($custoTotal$)

O modelo de previsão indica a classe de um objeto no próximo período de tempo. Para isso, o modelo é ajustado levando em consideração os padrões de acesso mais recentes dos objetos. Especificamente, organizamos o número de acessos aos objetos por unidade de tempo e os utilizamos sequencialmente em períodos fixos para treinar o modelo. Neste trabalho, adotamos unidades semanais e períodos mensais, seguindo outros estudos da literatura [Irie et al. 2018, Hsu et al. 2018]. Diferentes configurações podem ser definidas, dependendo da disponibilidade dos dados e das características dos objetos a serem previstos. Usamos os dois períodos mais recentes para treinar os modelos de previsão. Essa estratégia pressupõe que podemos prever acessos futuros com base nas atividades mais recentes observadas em objetos, semelhante à estratégia também adotada por algoritmos online [Liu et al. 2019, Erradi and Mansouri 2020], que usamos como linha de base para analisar o desempenho do nosso modelo.

Consideramos uma janela deslizante de comprimento fixo para deslocar os períodos para treinar os modelos de previsão. Por exemplo, a Figura 2 apresenta as janelas usadas para cada iteração em um objeto. Há 24 semanas nessa figura, com 6 períodos para treinar o arquivo. A primeira linha mostra o primeiro período e o segundo período (cada um composto por 4 semanas) como entrada e rótulo para treinamento do modelo, respectivamente. Além disso, o segundo e terceiro períodos são utilizados, respectivamente, como entrada e rótulo para avaliação do modelo. Note que, o segundo período é utilizado tanto como saída para treinamento quanto como entrada para avaliação do modelo. A segunda linha é a próxima janela deslizante. Nesse sentido, todas as janelas de tempo são deslocadas para o próximo período. Assim, toda previsão feita é baseada apenas no treinamento realizado no período anterior, desconsiderando os dados anteriores. Este processo se repete até que o último ponto seja usado como rótulo de avaliação.

De acordo com a metodologia de janela deslizante acima, o modelo prevê se o objeto deve pertencer à classe frequente ou infrequente. Considerando os custos cobrados pelo provedor de armazenamento em nuvem mostrados na Tabela 1, objetos com pelo menos uma solicitação de acessos no período devem ser classificados como frequentes.

A Figura 3 ilustra, no lado esquerdo, como o modelo é construído. Na linha horizontal mostramos unidades de tempo e períodos, enquanto na linha vertical mostramos n objetos armazenados na nuvem como a classe padrão, ou seja, objetos acessados com

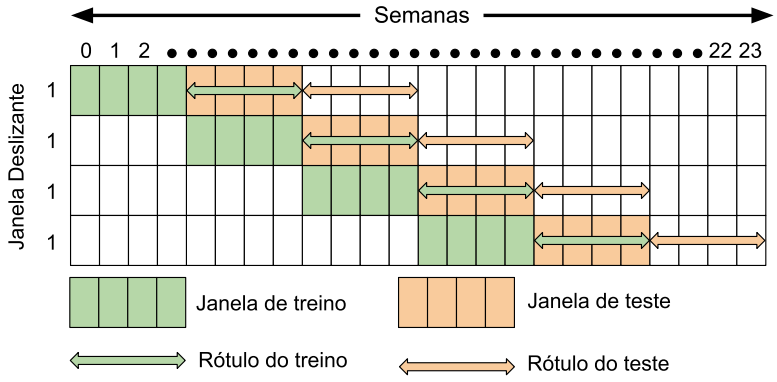

Figura 2. Janela deslizante de treino e avaliação em um único objeto

frequência. Retângulos verdes e retângulos com setas verdes representam períodos usados para treinar o modelo. O primeiro é o conjunto de treinamento que consiste em quatro pontos de unidade e o último é o conjunto de rótulos que representa um estado de classe binária, por exemplo, objeto acessado com frequência ou não. Assim, um modelo de previsão M é construído com base nos acessos aos objetos mais recentes que são processados nos períodos de conjunto de treinamento e conjunto de rótulos.

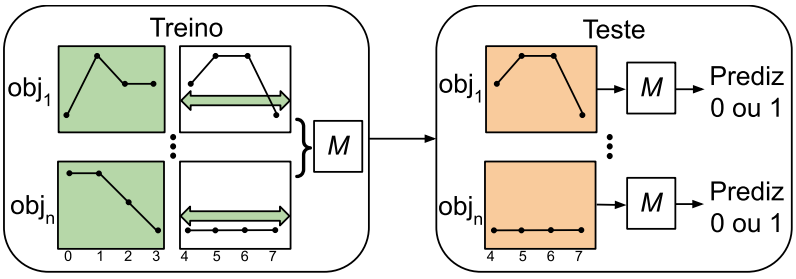

Figura 3. Nos períodos de treinamento (entrada e rótulo), um modelo *M* é construído baseado nos mais recentes acessos de *n* objetos armazenados na nuvem. Na fase de teste (entrada e rótulo), o modelo *M* é usado para prever qual classe cada um dos *n* objetos deve estar, sendo 1 a classe frequente e 0 a classe infrequente.

A Figura 3 também ilustra, no lado direito, a aplicação (ou teste) do modelo treinado M. Os acessos a objetos no próximo período (retângulo laranja) agora são usados como entrada do modelo. Pode-se notar que uma predição de classe tem um valor binário que é verdadeiro ou falso respectivamente para indicar um objeto frequente ou infrequente. A classe de objeto em nosso modelo é predefinida pelo limite para o número de acessos por período, mostrado na Equação 2. Abaixo desse limite, um objeto é classificado como infrequente e deve ser armazenado na camada oferecida pelos provedores de nuvem especificamente para esse padrão de acesso; caso contrário, esse objeto é classificado como frequente e deve ser mantido na camada de armazenamento padrão.

4. Avaliação

4.1. Configuração do Modelo

Como descrito anteriormente, o modelo proposto pode usar diferentes métodos para realizar a predição de classes de objetos. Neste trabalho, propomos modelos de aprendizado de máquina supervisionados para prever acessos futuros a objetos de forma binária. Usamos a estrutura de preços do armazenamento em nuvem mostrada na Tabela 1, que leva ao limite de número de acessos igual a uma solicitação aos objetos (ou seja, $\hat{y}_i \approx 1$). Portanto, basta que o modelo preveja se os objetos terão acesso zero ou maior que zero no período seguinte, o que significa uma classificação binária dos objetos para infrequentes ou frequentes, respectivamente.

Avaliamos onze modelos de aprendizado de máquina e o algoritmo *online* proposto em [Liu et al. 2019]. Todos esses métodos são representados por abreviações por questão de espaço: algoritmo *online* (ONL), Random Forest (RF), Linear Regression (LR), Stacking (STK), k-Nearest Neighbors (KNN), Decision Tree (DT), Support Vector Machine usando kernels RBF (SVM-R) e Sigmoid (SVM-S) e algoritmos de comitês como Soft Voting Standard (SV), Pondered (SV-P) e Grid (SV-gd) e Hard Voting (HV). Os modelos de aprendizado de máquina são implementados com a biblioteca scikit learn.[4] Por sua vez, o ONL monitora objetos dentro de um período de janela fixo de um mês e usa uma solicitação para o limite de número de acessos, o que faz com que ele se comporte de maneira semelhante às opções de classificação automática oferecidas por alguns provedores de armazenamento em nuvem.[5]

Métodos de pré-processamento de dados foram usados para lidar com classes desequilibradas no aprendizado de máquina. Um conjunto de dados é desequilibrado se as classes contidas nele não forem representadas de forma aproximadamente igual. Muitas vezes, os conjuntos de dados reais são predominantemente compostos de exemplos "normais" com apenas uma pequena parcela de "anormais". Para cada modelo, testamos os métodos *undersampling, oversampling* e SMOTE (Synthetic Minority Oversampling Technique). O SMOTE é uma abordagem de superamostragem na qual a classe minoritária é superamostrada criando exemplos "sintéticos" em vez de utilizar substituição [Chawla et al. 2002] .

A Tabela 2 mostra a soma dos custos de armazenamento de objetos com base na Equação 1 e a classe prevista pelos modelos com os métodos de pré-processamento. O SMOTE mostra o melhor resultado em 11 dos 22 testes. Portanto, estabelecemos o SMOTE para pré-processar dados em todos os modelos de aprendizado de máquina.

4.2. Base de dados

Usamos traços reais de acessos a dados armazenados no Dropbox,[6] coletados em estudo anterior [Gonçalves et al. 2016] por meio de dois pontos de presença (PoPs) de provedores de serviço de internet residenciais, denominados PoP-1 e PoP-2. Os traços registram os acessos dos usuários aos seus dados por meio desses PoPs, onde cada dado é um

[4]https://scikit-learn.org/stable/

[5]Por exemplo, O Amazon S3 fornece a opção *Intelligent-Tiering* para monitoramento e classificação automática de objetos com taxas extras por grupo de objetos.

[6]https://www.dropbox.com/

Tabela 2. Valores obtidos utilizando undersampling, oversampling e SMOTE para os classificadores considerados aqui.

PoP	Modelo	Undersampling	Oversampling	SMOTE
PoP-1	ONL	523,856328	523,856328	523,856328
	RF	451,957323	502,696826	464,639682
	LR	468,178918	529,709001	526,744467
	STK	396,070133	399,493303	392,263804
	KNN	422,124369	415,281970	410,609794
	DCT	522,760379	593,039376	522,118536
	SVM-R	391,622417	393,401642	390,383858
	SVM-S	491,385842	438,289748	438,815302
	SV	402,876414	404,757094	402,173421
	SV-P	401,888822	400,987508	398,890926
	SV-gd	391,664293	391,370603	**390,048911**
	HV	417,532004	418,557001	409,192609
PoP-2	ONL	512,195014	512,195014	512,195014
	RF	539,895228	549,678639	549,391301
	LR	558,028198	548,645781	553,366700
	STK	462,030013	485,212796	477,363645
	KNN	493,483047	487,524887	483,216476
	DCT	566,987820	571,733582	584,143142
	SVM-R	444,025187	450,974099	447,789093
	SVM-S	555,446274	553,081208	553,563437
	SV	480,946115	481,999434	474,952144
	SV-p	463,552099	479,831967	471,155600
	SV-gd	442,104576	451,051115	**439,724474**
	HV	509,398703	508,371621	512,842312

Dropbox *namespace*, ou seja, uma estrutura utilizada neste serviço para identificar de forma exclusiva um arquivo de usuário (documento, áudio, imagem) ou um diretório. Portanto, assumimos que cada *namespace* é um objeto que o usuário acessa no serviço de armazenamento em nuvem.

Os acessos dos usuários são representados por identificadores anônimos (ID), que não fornecem pistas sobre a identidade dos usuários ou o conteúdo armazenado no serviço, mas nos permitem analisar o padrão de acesso aos dados no Dropbox. As informações sobre os acessos dos usuários que usamos para nossa análise incluem o carimbo de data/hora de acesso, o ID do objeto e uma estimativa do volume do objeto. Esse volume é a soma de bytes em *uploads* e *downloads* por objeto, representando uma estimativa conservadora para cada objeto no conjunto de dados.

Usamos esses dados para construir séries temporais para cada objeto, onde as unidades de tempo representam o número de acessos por semana. Além disso, estimamos o volume desses objetos em cada unidade de tempo. No total, obtivemos 32 semanas para PoP-1 e 48 semanas para PoP-2, o que nos forneceu 8 e 12 períodos para análise em cada conjunto de dados, respectivamente, considerando períodos de quatro semanas do modelo conforme descrito na Seção 3.

4.3. Métricas

Usamos seis métricas para avaliar o desempenho dos modelos de aprendizado de máquina. Precisão indica a porcentagem de respostas corretas (verdadeiros positivos e verdadeiros negativos) sobre o número total de previsões feitas. O F_1-score é calculado pela média harmônica da precisão do modelo (fração de acertos entre as previsões positivas) e a revocação do modelo (fração de acertos entre a classe positiva), se os acertos do mo-

delo ocorrem simplesmente porque são tendenciosos pela classe mais representativa. O F_1-score é especialmente útil para casos de classes desequilibradas, ou seja, um F_1-score baixo indica se os acertos do modelo ocorrem porque é enviesado para a classe mais representativa. Quando precisamos considerar a revocação sendo duas vezes mais importante que a precisão, F_2-score pode ser usado. Nesse caso, o custo de um falso positivo não é o mesmo que o custo de um falso negativo. A AUC-ROC (*Area Under the Curve - Receiver Operating Characteristics*), por sua vez, representa o percentual da área ocupada pela curva ROC (variação da taxa de verdadeiros positivos com falsos positivos), e quanto maior esse percentual, melhor o modelo distingue as classes. Porcentagens próximas a 50% indicam que o modelo não é melhor do que uma previsão aleatória ou uma previsão fixa em uma única classe.

Para avaliar a economia com a redução do custo de armazenamento, usamos a métrica rcs [Gonçalves et al. 2016], abreviação de economia de custo relativa, dada por:

$$rcs = \sum_{t=1}^{T} \frac{custo\_padrão_t - custo\_do\_modelo_t}{custo\_padrão_t}, \tag{3}$$

onde t é o período atual, T é o número de períodos a serem avaliados, $custo\_padrão$ é o custo do serviço com todos os objetos na classe frequente (nenhuma abordagem de redução de custo é usada) e $custo\_do\_modelo$ é o custo do serviço usando nosso modelo. Ambos os custos são calculados para todos os objetos no período atual.

4.4. Resultados

Conduzimos as simulações com base nos traços PoP-1 e PoP-2 para representar a aplicação do modelo proposto de forma realista, conforme mostrado na Figura 2.

A Tabela 3 apresenta os resultados em termos de precisão, revocação, pontuações F_1 e F_2 e AUC-ROC para o desempenho de diferentes modelos de aprendizado de máquina no modelo. Por sua vez, rcs mostra as respectivas economias de custo de cada modelo em comparação com a opção de manter os objetos sempre armazenados na classe frequente, ou seja, a camada de armazenamento padrão. Primeiramente, notamos que todos os modelos avaliados levam a uma economia dada por rcs de 33% a 53%. Consideramos tal economia relevante para usuários de armazenamento em nuvem, pois é extraída de objetos ativos que podem ser acessados a qualquer momento pelo usuário com a mesma latência na classe frequente ou não frequente, conforme definido pelos contratos de nível de serviço dos provedores. Portanto, há um potencial de economia sem o risco de reduzir a qualidade do serviço.

Pode-se observar na Tabela 3 que, os comitês possuem alto desempenho, com Soft Voting grid (SV-gd) ligeiramente superior às versões ponderada e padrão. Considerando rcs, o SVM-R apresenta desempenho semelhante ao melhor resultado em ambos os conjuntos de dados, PoP-1 e PoP-2. Especificamente, sua precisão e AUC-ROC estão acima de 73% indicando que eles são capazes de distinguir classes corretamente para a maioria dos objetos. Além disso, sua pontuação F_1 é maior que 70%, indicando que o desequilíbrio de objetos nas classes frequentes e infrequentes, o que é natural ao longo do tempo em serviços de armazenamento, não interfere nas previsões de objetos na classe minoritária (frequente).

Tabela 3. PoP-1 e PoP-2 - Valores obtidos usando SMOTE para os classificadores aqui considerados.

PoP	Modelo	Acurácia	Precisão	Revocação	F_1-score	F_2-score	AUC-ROC	rcs
PoP-1	ONL	0,52877 (0,0160)	0,46703 (0,0767)	0,54061 (0,1156)	0,49890 (0,0868)	0,52246 (0,1015)	0,52029 (0,0058)	0,413434
	RF	0,74851 (0,0377)	0,77423 (0,0719)	0,63109 (0,1156)	0,68521 (0,0679)	0,65026 (0,0990)	0,74050 (0,0350)	0,370837
	LR	0,69862 (0,0655)	**0,85780 (0,0374)**	0,39811 (0,0697)	0,53984 (0,0633)	0,44452 (0,0695)	0,67245 (0,0313)	0,366284
	STK	0,75387 (0,0317)	0,77185 (0,0712)	0,64939 (0,1051)	0,69665 (0,0580)	0,66622 (0,0883)	0,74707 (0,0305)	0,453312
	KNN	0,74428 (0,0298)	0,74576 (0,0649)	0,65736 (0,0645)	0,69453 (0,0272)	0,67093 (0,0483)	0,73824 (0,0197)	0,446620
	DCT	0,73721 (0,0385)	0,77578 (0,0759)	0,59418 (0,1145)	0,66214 (0,0713)	0,61825 (0,1001)	0,72669 (0,0340)	0,331039
	SVM-R	0,75721 (0,0305)	0,77055 (0,0733)	0,66395 (0,1046)	0,70466 (0,0517)	0,67824 (0,0846)	0,75129 (0,0281)	0,487192
	SVM-S	0,73513 (0,0303)	0,74666 (0,0864)	0,64500 (0,1155)	0,68048 (0,0368)	0,65636 (0,0835)	0,73319 (0,0273)	0,366059
	SV	0,75629 (0,0281)	0,76343 (0,0615)	0,66420 (0,0676)	0,70638 (0,0317)	0,67969 (0,0522)	0,74998 (0,0223)	0,456084
	SV-P	0,75764 (0,0270)	0,76222 (0,0614)	0,67035 (0,0650)	**0,70953 (0,0295)**	**0,68476 (0,0495)**	0,75183 (0,0212)	0,460432
	SV-gd	**0,75843 (0,0293)**	0,77002 (0,0754)	**0,66777 (0,1062)**	0,70646 (0,0542)	0,68129 (0,0864)	**0,75273 (0,0289)**	**0,496427**
	HV	0,74905 (0,0377)	0,78035 (0,0727)	0,62613 (0,1125)	0,68476 (0,0612)	0,64687 (0,0945)	0,74124 (0,0329)	0,412693
PoP-2	ONL	0,54204 (0,0234)	0,47996 (0,1180)	0,49988 (0,1205)	0,48621 (0,1083)	0,49348 (0,1131)	0,51985 (0,0071)	0,376611
	RF	0,74461 (0,0918)	0,76749 (0,0688)	0,66099 (0,1456)	0,69508 (0,0869)	0,67217 (0,1252)	0,74505 (0,0551)	0,447079
	LR	0,71334 (0,1180)	**0,88665 (0,0308)**	0,44610 (0,1227)	0,58170 (0,1176)	0,49122 (0,1235)	0,69712 (0,0519)	0,373175
	STK	0,74493 (0,1130)	0,76826 (0,0766)	0,67699 (0,1771)	0,69679 (0,1250)	0,68251 (0,1592)	0,75305 (0,0647)	0,533206
	KNN	0,72800 (0,1143)	0,75189 (0,0883)	0,66978 (0,1622)	0,68794 (0,0922)	0,67368 (0,1343)	0,72205 (0,0976)	0,511375
	DCT	0,73046 (0,1001)	0,77786 (0,0767)	0,60411 (0,1579)	0,66064 (0,1108)	0,62357 (0,1433)	0,72966 (0,0544)	0,378679
	SVM-R	0,74919 (0,0945)	0,77090 (0,0784)	0,67976 (0,1571)	0,70386 (0,0933)	0,68680 (0,1346)	0,75489 (0,0555)	0,535444
	SVM-S	0,71740 (0,1102)	0,82542 (0,0768)	0,52468 (0,1989)	0,61015 (0,1463)	0,55281 (0,1836)	0,70777 (0,0706)	0,477810
	SV	0,74887 (0,0973)	0,78044 (0,0725)	0,65906 (0,1554)	0,69699 (0,0983)	0,67163 (0,1364)	0,75059 (0,0571)	0,521414
	SV-P	0,74948 (0,0966)	0,78004 (0,0729)	0,66142 (0,1543)	0,69839 (0,0973)	0,67364 (0,1353)	0,75169 (0,0569)	0,525320
	SV-gd	**0,75299 (0,0922)**	0,77225 (0,0803)	**0,68792 (0,1550)**	**0,70947 (0,0910)**	**0,69400 (0,1323)**	**0,75852 (0,0529)**	**0,535842**
	HV	0,74478 (0,1098)	0,80589 (0,0650)	0,61498 (0,1655)	0,67759 (0,1198)	0,63668 (0,1516)	0,74411 (0,0607)	0,513061

Quanto à economia de custos de armazenamento, no PoP-2, nosso melhor classificador alcança os maiores ganhos de economia com rcs acima de 53%, enquanto no PoP-1, rcs chega a 49%. Essa diferença se deve a uma característica do PoP-2, ou seja, a maioria dos objetos atinge a classe infrequente mais cedo. Além disso, os modelos obtiveram a revocação mais expressiva em relação à precisão no PoP-2, o que leva a menos falsos negativos. A redução desse tipo de erro é importante porque tem um impacto relevante no custo, ou seja, operações e requisições a um objeto da classe infrequente erroneamente podem fazer com que o custo final do objeto seja maior do que o custo de armazená-lo na classe frequente. Além disso, o custo desse erro aumenta proporcionalmente ao volume do objeto. Para lidar com essa questão, equilibramos igualmente as classes em treinamento, o que contribuiu para aumentar o número de acertos para a classe minoritária (frequente), reduzindo assim os falsos negativos.

Enquanto o SV-gd é um comitê, que utiliza mais recursos computacionais, o SVM-R é um SVM com kernel RBF. O SVM-R se destaca por ser um classificador simples e preciso, pois os resultados mostram que este modelo alcança a segunda melhor precisão e ganho de rcs em PoP-1 e PoP-2. Considerando a pequena diferença no ganho de rcs em relação ao SV-gd, o SVM-R pode ser apontado como um classificador eficiente também elegível ao modelo.

Finalmente, vale a pena notar que tanto o SV-gd quanto o SVM-R superam o algoritmo online (ONL), que é nossa linha de base de avaliação. Comparando os resultados de rcs de SV-gd e SVM-R com ONL, eles superam este último em 8,29% e 15,92% mais economia de custos, respectivamente.

5. Conclusões e trabalhos futuros

Neste trabalho, propomos uma estrutura para prever classes de objetos em serviços de armazenamento em nuvem hierárquicos para reduzir os custos de armazenamento. Avali-

amos essa estrutura usando rastreamentos de acesso de usuários reais coletados do Dropbox. Os resultados mostram potencial de economia com redução de custos que pode chegar a 53% sem risco de perda de qualidade para o usuário ao explorar as classes frequentes e infrequentes. Esses benefícios podem ser obtidos graças a modelos de previsão incorporados em um modelo para processamento automático dos acessos mais recentes de objetos armazenados, como apresentamos aqui.

Os modelos de comitês, como o soft e hard voting obtiveram os melhores resultados em rcs. Considerando as demais métricas avaliadas, a AUC-ROC é a que mais se aproxima dos resultados em rcs. Assim, AUC-ROC é uma boa métrica potencial para problemas onde o cálculo rcs não está disponível.

Trabalhos futuros incluem a avaliação do modelo proposto com mais classes, incluindo objetos inativos, e mecanismos de incentivo para compensar a perda de qualidade de serviço para o usuário com a classe de armazenamento. Também é possível analisar diferentes tamanhos de períodos e o impacto de modificar esses valores. Planejamos encontrar melhorias de desempenho dos modelos de previsão com configurações otimizadas para classificadores e assim reduzir o custo de armazenamento. Outra possibilidade de trabalho futuro inclui realizar experimentos com diferentes bases de dados para assim analisar os ganhos do método em bases com diferentes características. Uma extensão adicional deste trabalho é usar métodos de aprendizado profundo para tentar alcançar melhores resultados.

Referências

Bocchi, E., Drago, I., and Mellia, M. (2015). Personal Cloud Storage: Usage, Performance and Impact of Terminals. In *Proc. of the IEEE CloudNet*.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Cisco (2019). Cisco Global Cloud Index: Forecast and Methodology, 2016–2021 White Paper. Disponível em `https://www.cisco.com` - Document ID 1513879861264127.

Erradi, A. and Mansouri, Y. (2020). Online cost optimization algorithms for tiered cloud storage services. *Journal of Systems and Software*, 160:110457.

Gonçalves, G., Drago, I., Silva, A. P. C., Vieira, A. B., and Almeida, J. M. (2016). The impact of content sharing on cloud storage bandwidth consumption. *IEEE Internet Computing*, 20(4):26–35.

Gracia-Tinedo, R., García-López, P., Gómez, A., and Illana, A. (2016). Understanding data sharing in private personal clouds. In *Proc. of the IEEE CLOUD*.

Hsu, Y., Irie, R., Murata, S., and Matsuoka, M. (2018). A novel automated cloud storage tiering system through hot-cold data classification. In *Proc. of the IEEE CLOUD*.

Irie, R., Murata, S., Hsu, Y., and Matsuoka, M. (2018). A novel automated tiered storage architecture for achieving both cost saving and qoe. In *Proc. of the IEEE SC2*.

Kaushik, R. T. and Bhandarkar, M. (2010). Greenhdfs: towards an energy-conserving, storage-efficient, hybrid hadoop compute cluster. In *Proc. of the USENIX*.

Liu, M., Pan, L., and Liu, S. (2019). To transfer or not: An online cost optimization algorithm for using two-tier storage-as-a-service clouds. *IEEE Access*, 7:94263–94275.

Liu, M., Pan, L., and Liu, S. (2021). Keep hot or go cold: A randomized online migration algorithm for cost optimization in staas clouds. *IEEE Transactions on Network and Service Management*.

Mesnier, M., Ganger, G. R., and Riedel, E. (2003). Object-based storage. *IEEE Communications Magazine*, 41(8):84–90.

Ribeiro, S., Gonçalves, G., Silva, F., Vieira, A., and Almeida, J. (2020). 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*, pages 85–96, Porto Alegre, RS, Brasil. SBC.

Subramanian, M., Lloyd, W., Roy, S., Hill, C., Lin, E., Liu, W., Pan, S., Shankar, S., Viswanathan, S., Tang, L., and Kumar, S. (2014). f4: Facebook's warm blob storage system. In *Proc. of the OSDI*.