

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
DEPARTAMENTO DE COMPUTAÇÃO

FABIO AUGUSTO SOUZA DO NASCIMENTO

Modelos de recomendação sensível ao contexto em ambientes de comunicação
aumentativa alternativa: naive bayes, redes neurais e aprendizagem federada

RECIFE
2025

FABIO AUGUSTO SOUZA DO NASCIMENTO

Modelos de recomendação sensível ao contexto em ambientes de comunicação aumentativa alternativa: naive bayes, redes neurais e aprendizagem federada

Trabalho de Conclusão de Curso apresentado ao Corpo Docente do Departamento de Computação da Universidade Federal Rural de Pernambuco, como parte dos requisitos necessários à obtenção do título de Bacharel em Ciência da Computação.

Orientador(a):

André Câmara Alves do Nascimento

Recife, PE

2025

Dados Internacionais de Catalogação na Publicação
Sistema Integrado de Bibliotecas da UFRPE
Bibliotecário(a): Suely Manzi – CRB-4 809

N244m Nascimento, Fabio Augusto Souza do.
Modelos de recomendação sensível ao contexto em ambientes de comunicação aumentativa alternativa: naive bayes, redes neurais e aprendizagem federada / Fabio Augusto Souza do Nascimento. – Recife, 2025.
38 f.; il.

Orientador(a): André Câmara Alves do Nascimento.

Trabalho de Conclusão de Curso (Graduação) – Universidade Federal Rural de Pernambuco, Bacharelado em Ciência da Computação, Recife, BR-PE, 2025.

Inclui referências.

1. Aprendizado do computador. 2. Algoritmos computacionais. 3. Redes neurais (Computação). 4. Inteligência artificial 5. Computação evolutiva. I. Nascimento, André Câmara Alves do, orient. II. Título

CDD 004

FABIO AUGUSTO SOUZA DO NASCIMENTO

Modelos de recomendação sensível ao contexto em ambientes de comunicação aumentativa alternativa: naive bayes, redes neurais e aprendizagem federada

Trabalho de Conclusão de Curso apresentado ao Corpo Docente do Departamento de Computação da Universidade Federal Rural de Pernambuco, como parte dos requisitos necessários à obtenção do título de Bacharel em Ciência da Computação

Aprovado em __ de março de 2025, com nota __, __ (valor, valor), pela banca examinadora.

BANCA EXAMINADORA

Prof. Dr. André Câmara Alves do Nascimento
UFRPE

Prof. Dr. Ruan Vasconcelos Bezerra Carvalho
UFRPE

RESUMO

A inteligência artificial (IA) está presente em diversos setores da sociedade, promovendo constantes avanços tecnológicos e sociais. Um dos subconjuntos da IA é a aprendizagem de máquina (AM), que possibilita aos computadores aprenderem e evoluírem com base em dados. O conhecimento obtido por meio desse processo auxilia em diferentes áreas, desde a análise e interpretação de informações até a usabilidade de aplicativos. Compreender o funcionamento e as possibilidades dessa ferramenta pode ser um ativo valioso. Entre as múltiplas aplicações possíveis, a AM desempenha um papel importante na comunicação aumentativa e alternativa (CAA), auxiliando indivíduos que apresentam dificuldades nessas interações por meio de aplicativos. Essas aplicações estão em constante evolução e, com as inovações tecnológicas, possibilitam o desenvolvimento de sistemas capazes de compreender o ambiente em que uma pessoa está inserida e oferecer recomendações personalizadas aos usuários. Este trabalho investiga o uso dos Sistemas de Recomendação Sensível ao Contexto (SRSC) em aplicações da CAA, considerando as características individuais dos usuários com base em seus dados e contexto. São utilizados modelos de IA, tais como *Naïve Bayes* (NB), Redes Neurais Artificiais (RNA) e Aprendizagem Federada (AF), para comparar diferentes abordagens e avaliar sua capacidade de fornecer resultados relevantes. A partir dos experimentos realizados, foi possível verificar que modelos personalizados demonstram melhor desempenho em relação a abordagens globais, oferecendo recomendações mais relevantes aos usuários finais. Nesse sentido, a personalização e o uso de variáveis contextuais podem melhorar significativamente a experiência de pessoas que dependem de CAA, aumentando a agilidade e a assertividade da comunicação.

Palavras-chave: Aprendizagem de Máquina, Comunicação Aumentativa e Alternativa, Sistemas de Recomendação Sensível ao Contexto.

ABSTRACT

Artificial intelligence (AI) is present in various sectors of society, driving continuous technological and social advancements. One of AI's subsets is machine learning (ML), which enables computers to learn and evolve based on data. The knowledge obtained through this process supports different fields, from information analysis and interpretation to application usability. Understanding the functioning and potential of this tool can be an asset. Among its multiple applications, ML plays a significant role in augmentative and alternative communication (AAC), assisting individuals who face difficulties in communication through specialized applications. These applications are constantly evolving and, with technological innovations, enable the development of systems capable of understanding the environment in which a person is situated and providing personalized recommendations to users. This study investigates the use of Context-Aware Recommender Systems (CARS) in AAC applications, considering users' individual characteristics based on their data and context. AI models such as Naïve Bayes (NB), Artificial Neural Networks (ANN), and Federated Learning (FL) are employed to compare different approaches and assess their ability to provide relevant results. The experiments conducted demonstrate that personalized models outperform global approaches, offering more relevant recommendations to end users. In this regard, personalization and the use of contextual variables can significantly enhance the experience of individuals relying on AAC, improving communication speed and accuracy.

Keywords: Machine Learning, Augmentative and Alternative Communication, Context-Aware Recommendation Systems.

LISTA DE FIGURAS

Figura 1 – Exemplo de sistema auxiliado tradicional	5
Figura 2 – Exemplo de sistema auxiliado com recomendação	6
Figura 3 – Exemplo de rede neural artificial	9
Figura 4 – Exemplo da arquitetura duas torres	9
Figura 5 – Exemplo de arquitetura de ranking	10
Figura 6 – Exemplo de arquitetura federada	11
Figura 7 – Fluxo de processamento de ML	14
Figura 8 – Distribuição da quantidade de cartões por usuário	17
Figura 9 – Distribuição da quantidade de localizações distintas por usuário	17
Figura 10 – Exemplo de separação dos dados na validação cruzada	18
Figura 11 – Proporção entre treinamento e teste por usuário	19
Figura 12 – Resultado do modelo Naïve Bayes	23
Figura 13 – Resultado do modelo RNA	24

LISTA DE TABELAS

Tabela 1 – Modelos e valores dos parâmetros utilizados	21
Tabela 2 – Recomendação do modelo NB	23

LISTA DE ABREVIATURAS E SIGLAS

AAC	Augmentative and Alternative Communication
AF	Aprendizagem Federada
AM	Aprendizagem de Máquina
ANN	Artificial Neural Network
CAA	Comunicação Aumentativa Alternativa
CARS	Context-aware Recommendation System
FL	Federated Learning
IA	Inteligência Artificial
LLM	Large Language Model
NB	Naive Bayes
SR	Sistema de Recomendação
SRSC	Sistema de Recomendação Sensível ao Contexto
TF	TensorFlow

SUMÁRIO

1	INTRODUÇÃO	2
1.1	Objetivos.....	3
1.2	Estrutura do Trabalho.....	4
2	FUNDAMENTAÇÃO TERÓRICA.....	5
2.1	Comunicação Aumentativa Alternativa.....	5
2.2	Sistema de Recomendação	6
2.3	Sistema de Recomendação Sensível ao Contexto	7
2.3.1	Sistema de Recomendação Baseado em NB.....	8
2.3.2	Sistema de Recomendação Baseado em Redes Neurais.....	9
2.3.3	Sistema de Recomendação Baseado em AF	10
2.4	Trabalhos Relacionados	11
3	METODOLOGIA	14
3.1	Conjunto de Dados	14
3.1.1	Preparação e Transformação dos Dados.....	15
3.1.2	Filtragem e Visualização dos Dados	16
3.1.3	Separação dos Dados.....	17
3.2	MODELOS	19
3.2.1	<i>Naïve Bayes</i>	19
3.2.2	Rede Neural Artificial.....	20
3.2.3	Aprendizagem Federada	20
3.2.4	Parâmetros e Métricas de Avaliação.....	21
4	Experimentos e Resultados	22
4.1	RESULTADOS	22
5	CONCLUSÃO.....	25
5.1	Limitações	25
5.2	Trabalhos Futuros.....	26
	REFERÊNCIAS BIBLIOGRÁFICAS	27

1 INTRODUÇÃO

A comunicação é uma forma básica de interação entre os seres vivos. Ela pode ser realizada de diversas maneiras, por meio da fala, sons, gestos, símbolos e escrita. Esse conjunto de habilidades possibilita a troca de ideias, sentimentos e informações [1]. Embora a maioria das pessoas não enfrente dificuldades ao se comunicar, algumas apresentam limitações nessa capacidade devido a deficiências, físicas, cognitivas ou sensoriais.

Essas limitações podem impactar significativamente o processo comunicativo tornando-o desafiador. As dificuldades podem ser causadas por fatores adquiridos ao longo da vida, como doenças e acidentes ou presentes desde o nascimento. Independente da origem, a dificuldade de comunicação pode levar ao isolamento social e impactar a qualidade de vida de uma pessoa [2].

A comunicação aumentativa e alternativa surge como forma de auxiliar indivíduos que vivem nessa realidade, ampliando suas possibilidades de interação. A CAA compreende um conjunto de estratégias, métodos e ferramentas que atuam na redução das barreiras comunicativas e no aprimoramento cognitivo dos usuários, promovendo uma melhor interação e inclusão social [2]. Além disso, estimula a elaboração de mensagens e auxilia na tomada de decisão sobre os recursos de comunicação disponíveis para o indivíduo.

Um estudo realizado em [3] demonstra que o uso da CAA gera melhorias substanciais na produção verbal espontânea. Em 89% dos casos avaliados, houve aumento na capacidade de comunicação após intervenções com CAA. Esses estudos utilizaram estratégias baseadas em sinais manuais e no sistema de comunicação por troca de figuras. Os ganhos observados na comunicação verbal variaram desde incrementos moderados até expressivos, chegando a mais de 50 novas palavras ou frases produzidas espontaneamente pelos participantes [3].

O aprimoramento da CAA tem sido impulsionado pelo avanço de tecnologias, especialmente com o uso de inteligência artificial [4]. A utilização de uma aplicação de CAA por um indivíduo pode capturar os dados gerados por esse usuário durante sua utilização. Esses dados podem ser empregados em modelos de AM para identificar padrões e preferências desse indivíduo [4]. Dessa forma, o sistema torna-se capaz de sugerir recomendações ao usuário final sem que ele declare uma necessidade, baseando-se apenas nos dados fornecidos.

Para se alcançar um bom resultado em um sistema de recomendação (SR), são necessários tempo e estudo. Com a existência de diversos modelos [5] aplicáveis a um SR, encontrar uma configuração relevante é, por si só, uma tarefa desafiadora. Modelos tradicionais, como o *Naïve Bayes*, que busca soluções de forma probabilística, bem como modelos de redes neurais artificiais e aprendizagem federada, são alguns desses exemplos.

Embora os SR possam fornecer soluções diversas [6] [7], ainda há aspectos que podem ser explorados para aprofundar o conhecimento sobre o tema, como os SRSC. Os modelos sensíveis ao contexto consideram não apenas os dados fornecidos diretamente pelos usuários, mas também aqueles relacionados ao ambiente em que as interações ocorreram. Ao incorporar informações contextuais, os modelos podem proporcionar melhorias significativas [8], permitindo uma compreensão mais precisa das escolhas feitas pelos usuários, aumentando a qualidade das recomendações.

O Livox [9] é um aplicativo de CAA que emprega tecnologias de inteligência artificial e aprendizagem de máquinas, integradas a sistemas de recomendação. Em um trabalho anterior [10], foram utilizados dados do Livox para avaliar modelos baseados em RNA dentro de um SRSC. No entanto, essa abordagem se restringiu a modelos de recomendação globais. Considerar as preferências pessoais dos usuários pode aumentar a precisão das recomendações, tornando-as mais relevantes.

Considerando as possibilidades de modelos personalizados, ainda não exploradas no ambiente de sistemas de recomendações sensíveis ao contexto, e o impacto positivo que ferramentas de CAA mais precisas podem gerar para seus usuários [3], justifica-se a realização de estudos que avaliem abordagens personalizadas nesse contexto.

1.1 Objetivos

Esse estudo tem por objetivo explorar a aplicação de modelos como o NB, RNA e AF no contexto de CAA, utilizando um SRSC. Busca-se investigar a eficácia de modelos personalizados em comparação aos modelos globais, utilizando uma base de dados do aplicativo Livox. Analisando e comparando os resultados obtidos, avaliando a capacidade dos modelos em realizar recomendações relevantes aos usuários do aplicativo e contribuir para avanços no estado da arte.

1.2 Estrutura do Trabalho

O trabalho pós introdução, segue com os seguintes capítulos:

- Capítulo 2, aborda o referencial teórico e trabalhos recentes;
- Capítulo 3, demonstra a metodologia da pesquisa;
- Capítulo 4, apresenta os experimentos realizados e seus resultados;
- Capítulo 5, segue com as conclusões finais e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Comunicação Aumentativa Alternativa

Na CAA, existem dois principais grupos: os não auxiliados e os auxiliados. No primeiro grupo, o usuário consegue se comunicar diretamente por meio de línguas de sinais, linguagem corporal ou gestos. No segundo, o usuário utiliza ferramentas ou dispositivos, como pictogramas, fotos, sentenças prontas para aprimorar sua comunicação [2].

Os aparelhos de CAA são adaptáveis às diversas necessidades dos usuários e podem variar em relação ao grau de tecnologia envolvida [4]. Nesse sentido, a tecnologia desempenha um papel importante. Os dispositivos eletrônicos modernos oferecem novas possibilidades para esse segmento, permitindo maior acessibilidade e mobilidade desses aparelhos [4]. Além disso, avanços em conexões wireless, inteligência artificial e aprendizagem de máquina contribuem para aprimorar a precisão dos serviços prestados por essas ferramentas [11].

O Tobii Dynavox [12], o Eyegaze [13], o Proloquo2Go [14] e o brasileiro Livox são alguns dos produtos de CAA disponíveis no mercado. O Livox possibilita a comunicação por meio de pictogramas, que são imagens e textos organizados em cartões para auxiliar as interações do usuário. Esses pictogramas podem ser personalizados, permitindo a adição de novos cartões [9].

Em sistemas auxiliados tradicionais, o usuário se expressa por meio de uma sequência de opções ou botões. Esse processo aparentemente simples pode ser demorado e cansativo dependendo da mobilidade do usuário e quantidade de passos necessários para se comunicar. Dispositivos como o Livox, ao utilizarem pictogramas, conseguem melhorar tanto a visualização quanto a quantidade de interações necessárias, proporcionando uma comunicação mais ágil [9].

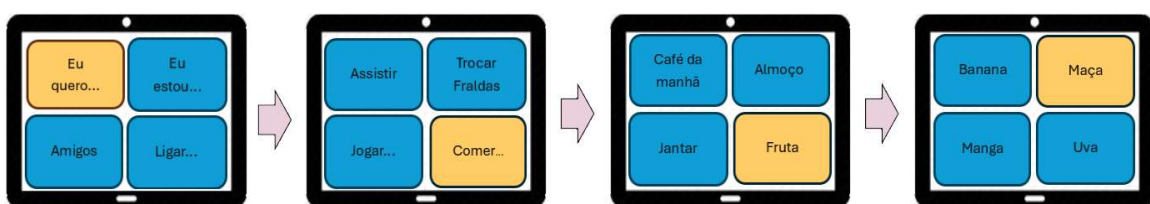


Figura 1: Exemplo de sistema auxiliado tradicional. Fonte: [15]

Na figura 1 é mostrado um exemplo da comunicação em um sistema tradicional que passa por diferentes quadros de opções, enquanto na figura 2 essa mesma comunicação tem uma melhoria na questão visual e redução na quantidade necessária de passos por utilizar um sistema de recomendação.

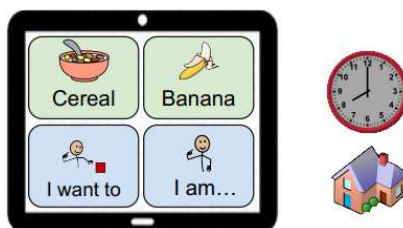


Figura 2: Exemplo de sistema auxiliado com recomendação. Fonte: [15]

Na figura 2 além do sistema auxiliado tradicional (azul), existem sugestões em verde que permitem pular toda a sequência de escolhas, indo direto à ação desejada.

2.2 Sistema de Recomendação

Os SR utilizam algoritmos para processar as escolhas feitas pelo usuário, armazenando um histórico de preferências. Com base em dados, o algoritmo analisa padrões de comportamento criando um perfil do indivíduo. A partir desse perfil, o sistema prevê itens de interesse e sugere recomendações [11]. Essa tecnologia oferece vantagens, como como redução no tempo de busca por informações ou sugestão de itens relevantes ao usuário [11].

Existem diferentes tipos de SR, com destaque para os sistemas baseados em conteúdo, filtragem colaborativa e híbridos. Os sistemas baseados em conteúdo utilizam as descrições dos itens e as escolhas prévias do usuário para construir um perfil a partir de suas avaliações. Isso permite que o algoritmo compare novos itens com os já avaliados e recomende aquele com melhor pontuação [16]. Já os sistemas que utilizam filtragem colaborativa analisam um volume maior de dados, comparando padrões de similaridade entre novos itens e as preferências do usuário para realizar recomendações [16]. O sistema híbrido combina ambas as abordagens, permitindo uma análise separada dos dados e a posterior união dos resultados [16].

Um SR pode ser impreciso quando utilizado por um novo usuário [17]. Sem informações prévias, o sistema pode gerar sugestões genéricas até que o perfil do

usuário seja enriquecido com novos dados, um problema conhecido como *cold start* [18].

2.3 Sistema de Recomendação Sensível ao Contexto

Como já visto, um sistema de recomendação sensível ao contexto permite que o algoritmo utilize não apenas os dados de entrada fornecidos pelo usuário, mas também as informações do ambiente em que o usuário está [19].

Contexto refere-se ao conjunto de circunstâncias ou condições em que ocorre um evento ou situação. Este conjunto pode incluir elementos do ambiente físico, social, cultural ou temporal, influenciando a compreensão e interpretação de um determinado fenômeno ou comportamento [3]. O contexto é essencial para atribuir significado a informações, ações ou eventos, pois fornece bases para que esses elementos sejam interpretados com a maior precisão.

Devido à diversidade de possibilidades em um contexto, existem várias sub definições, dependendo da área em que é aplicado [20]. Na computação, esse termo pode ser considerado como uma entidade representada por uma pessoa, lugar ou objeto relevante na interação entre usuário e sua aplicação, incluindo a própria aplicação [20].

O SRSC utiliza os mesmos princípios dos sistemas de recomendação tradicionais, com pequenas variações relacionadas ao contexto. As informações contextuais podem ser obtidas de forma explícita, implícita ou por inferência [21].

As informações explícitas são coletadas diretamente do usuário por meio de formulários ou perguntas específicas. As implícitas são extraídas do ambiente, sem necessidade de interação direta com o usuário, considerando variáveis como localização, horário, dia da semana ou estações do ano [22]. Na inferência, os dados são obtidos por dedução com base nos dados disponíveis [22].

Um exemplo prático desse processo é a identificação de mudanças nos hábitos de consumo em plataformas de *streaming*. Se um usuário que normalmente não assiste a programas infantis passa a consumi-los com frequência, o sistema pode inferir a presença de uma criança no ambiente. Com base nessa inferência, poderia sugerir recomendações voltadas a essa faixa etária ou até recomendar a criação de um perfil infantil.

2.3.1 Sistema de Recomendação Baseado em *Naïve Bayes*

A técnica baseada em *Bayes*, utiliza uma classificação probabilística, assumindo independência condicional entre as variáveis de entrada. Nesse modelo, cada atributo, incluindo as escolhas do usuário e o contexto, contribui de forma independente para a determinação da classe ou item final [23].

A fórmula do Teorema de *Bayes* é dada por:

$$P(A|B) = \frac{P(B|A)*P(A)}{P(B)} \quad (1)$$

Onde [24]:

- $P(A|B)$ representa a probabilidade posterior. Seria a probabilidade de um usuário escolher um determinado item considerando o contexto.
- $P(B|A)$ é a verossimilhança, indicando a probabilidade de um contexto ocorrer para um determinado item.
- $P(A)$ se refere a probabilidade prévia. A frequência com que o usuário escolhe A sem considerar o contexto.
- $P(B)$ é a evidência, corresponde a frequência com que o contexto ocorre no total das escolhas.

O modelo estima a probabilidade posterior da escolha de um item, com base nos dados, classificando-o de acordo com suas características e os associa às preferências do usuário [23]. Além disso, apresenta boa eficiência computacional e escalabilidade [17].

Com base nessas capacidades, o NB pode ser utilizado para construir modelos de recomendação híbridos [25], combinando a abordagem com a filtragem colaborativa. Essa integração aprimora a recomendação ao adicionar informações probabilísticas sobre os itens, associando padrões extraídos das preferências de múltiplos usuários. Uma das vantagens dessa abordagem é sua capacidade de lidar com conjuntos esparsos e auxiliar na fase inicial do sistema, mitigando o problema de *cold start* [25].

2.3.2 Sistema de Recomendação Baseado em Redes Neurais

Sistemas de recomendação baseado RNA utilizam de neurônios artificiais, organizados em camadas interconectadas por pesos ajustáveis. A camada de entrada recebe os dados, que são processados por camadas ocultas, onde padrões são extraídos e representações mais abstratas são criadas [26].

A camada de saída gera as recomendações com base nas ativações das camadas anteriores. O aprendizado ocorre por meio do ajuste iterativo dos pesos sinápticos, minimizando o erro entre as saídas previstas e os valores esperados, refinando a capacidade do modelo de gerar recomendações precisas [27].

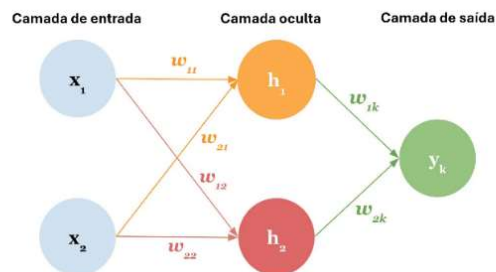


Figura 3: Exemplo de rede neural artificial. Fonte: editada em [28]

Redes neurais artificiais podem utilizar a arquitetura de duas torres para modelar separadamente as características dos usuários e dos itens recomendados. Cada torre consiste em uma rede neural que processa suas respectivas informações, gerando representações vetoriais [29]. Essas representações são então combinadas, e a similaridade entre usuário e item é calculada. Esse valor final é utilizado para classificar e recomendar os itens mais relevantes [29].

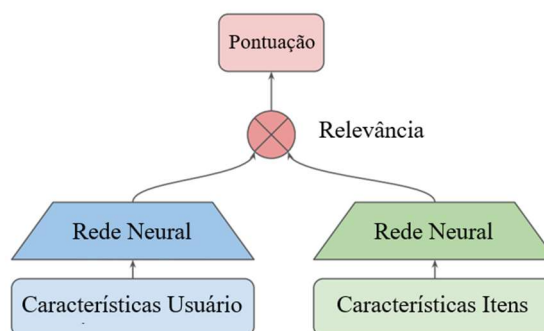


Figura 4: Exemplo da arquitetura duas torres. Fonte: editada em [29]

A arquitetura de *ranking* é outra abordagem que pode ser utilizada em modelos RNA. Nesse modelo os itens são ordenados de acordo com uma pontuação gerada pela rede neural [30]. Essa pontuação é obtida a partir da similaridade entre as representações vetoriais de usuários e itens, calculada por operações de produto escalar [31]. Em seguida, uma função de perda é aplicada para comparar a ordenação prevista com a real, ajustando os pesos da rede neural aprimorando o modelo [29].

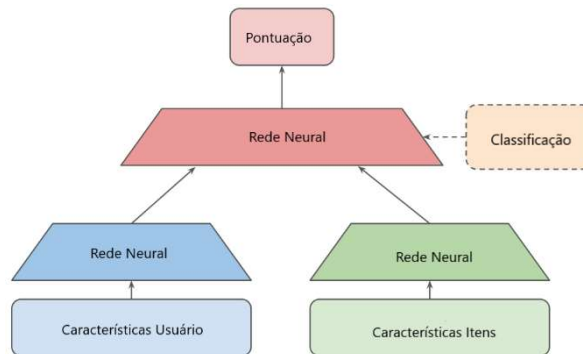


Figura 5: Exemplo de arquitetura de ranking. Fonte: editada em [29]

As RNA possibilitam identificar padrões mais complexos nas interações entre os usuários e itens. Abordagens baseadas em redes neurais profundas são capazes de capturar relações não lineares que modelos tradicionais não conseguem mapear [32], contribuindo para recomendações eficientes. Dependendo das abordagens utilizadas podem reduzir o impacto da esparsidade [32].

2.3.3 Sistema de Recomendação Baseado em Aprendizagem Federada

A aprendizagem federada é uma arquitetura que possibilita múltiplas partes realizarem o treinamento do algoritmo de forma colaborativa. No modelo padrão, um algoritmo global (servidor) é treinado e compartilhado com as múltiplas partes (ilhas) [33]. Essas ilhas possuem bases de dados locais e, ao receberem as informações do servidor, realizam um novo treinamento incorporando seus próprios dados [34]. Uma vez treinados, os modelos locais alimentam o modelo global no servidor, que agrega os novos dados e repete o processo [33].

A aprendizagem federada pode ser organizada dependendo da infraestrutura e dos modelos utilizados. Uma das abordagens mais comuns é o *FedAvg* (*Federated*

Averaging), onde cada ilha treina múltiplas rodadas localmente, e os pesos do modelo global são atualizados com as médias dos pesos dos modelos locais [33].

A aprendizagem federada pode ainda ser classificada em diferentes tipos com base na distribuição dos dados [34]:

- AF horizontal: Quando os dados possuem características similares, mas estão distribuídos entre diferentes representações locais. Muito comum em dispositivos móveis.
- AF vertical: Ocorre quando diferentes representações locais possuem informações sobre os mesmos indivíduos, mas com características distintas.
- AF por transferência: Quando não há sobreposição entre os dados das representações locais, tanto no espaço amostral quanto nas características. Nesse caso, é necessário utilizar técnicas de aprendizado por transferência para aprimorar o modelo.

No estudo realizado em [35] foi possível treinar o modelo localmente nos dispositivos dos usuários sem compartilhar de seus dados, apenas os parâmetros para o servidor. Esse processo também permitiu ao modelo aprender de forma descentralizada. Além proporcionar personalização sem comprometer a privacidade dos usuários [35]

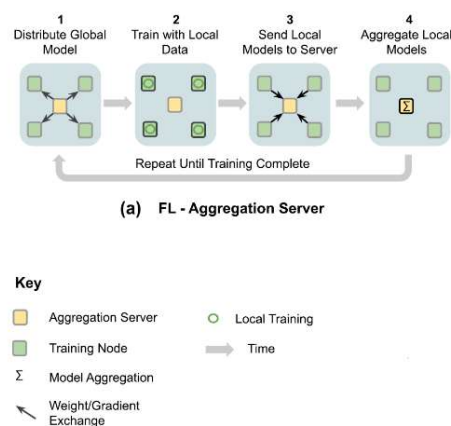


Figura 6: Exemplo de arquitetura federada. Fonte: editada de [36]

2.4 Trabalhos Relacionados

O trabalho apresentado em [15] explora o uso da inteligência artificial em CAA, utilizando o dispositivo Livox como ferramenta para auxiliar pessoas com dificuldades

comunicativas. O sistema propõe sugestões de pictogramas, buscando minimizar o tempo e esforço necessários nessa interação.

A utilização de redes neurais profundas realizada em [37] evidencia o potencial dessa abordagem como solução para sistemas de recomendação. Os autores combinaram *embeddings* e as múltiplas camadas de RNA para personalizar as recomendações. Como resultado o estudo demonstrou ganhos em precisão e eficiência computacional por meio do paralelismo.

O estudo feito em [38] apresenta um mapeamento sobre a predição de pictogramas em CAA, analisando diferentes abordagens computacionais utilizadas para prever pictogramas, como modelos estatísticos e RNA. Embora os métodos tenham sido avaliados por especialistas e por análises automáticas, não foi possível comprovar melhorias na comunicação dos usuários.

Trabalhos recentes [39] investigam o impacto de tecnologias assistivas na independência e participação dos indivíduos com deficiência. Destacando a CAA em SRSC e seu papel na comunicação de pessoas com dificuldades na fala. Foi realizada uma revisão de literatura para compreender os avanços, desafios e aplicações desses sistemas.

O trabalho apresentado em [10] aborda avaliações de RNA em SRSC com dados de dispositivos de CAA. Nesse estudo, foram utilizadas abordagens de recuperação (arquitetura de duas torres) e *ranking*. Ambos os modelos utilizaram o *TensorFlow* [40], uma ferramenta que facilita a criação de modelos de AM. Foi considerado, nesse estudo, representações de contexto como a localização, data e hora.

Os níveis de precisão obtidos na aplicação dos modelos não atingiram pontuação superior a 23,94%, e algumas limitações metodológicas podem ter contribuído para esse resultado. Os modelos desenvolvidos não consideraram a personalização por usuário, restringindo-se a um modelo global e generalista. Isso levou a recomendações pouco relevantes, uma vez que os itens preditos, não necessariamente correspondiam aos que o usuário realmente possuía.

Outro ponto que pode ter impactado os resultados foi a validação escolhida para esses modelos. A metodologia adotou a validação cruzada, que embora prática, pode não ter sido ideal para essa abordagem global. Ao separar os dados sem critérios é possível que informações de alguns usuários tenham ficado desproporcionais entre o treinamento e testes.

Podem ter ocorrido casos em que as informações de um usuário estivessem presentes apenas no conjunto de dados para treinamento ou só nos testes, o que traria impacto direto no treinamento e resultado das validações.

Explorar essas limitações com modelos personalizados, pode contribuir para melhoria da aplicação dos modelos já citados, incluindo os baseados em RNA. Nas pesquisas realizadas, não foi encontrado outro estudo que utilizasse modelos personalizados em base de dados do Livox especificamente no contexto de SRSC, reforçando o potencial do estudo.

3 METODOLOGIA

A utilização de algoritmos para o processamento de dados e a comprovação da hipótese que modelos personalizados podem ter um ganho na precisão e recomendação em relação a modelos globais, aliadas à realização de validações das informações, indicam o uso da metodologia quantitativa [41].

Os testes e avaliações foram realizados para a aplicação dos modelos utilizando os algoritmos de NB, redes neurais artificiais e aprendizagem federada.

3.1 Conjunto de Dados

O conjunto de dados foi extraído da base do aplicativo Livox, que possui informações de contexto, passíveis de análise pelos modelos. As informações dos usuários permaneceram privadas e não foram divulgadas, garantindo o respeito à privacidade e conformidade com a lei de proteção de dados LGPD 13.709 [42].

A base de dados do Livox utilizada no projeto, contém 982.754 linhas. Esses dados passaram por um processo de tratamento durante o pré-processamento, visando obter base viável de informações compreensíveis e utilizáveis no treinamento dos modelos. Essas adaptações são essenciais para garantir que os algoritmos de aprendizado de máquina operem corretamente e forneçam resultados confiáveis [43].

O projeto seguiu o fluxo de processamento em [44], iniciando com a coleta e preparação dos dados, passando pelo treinamento dos modelos e, em seguida, pela avaliação e otimização.

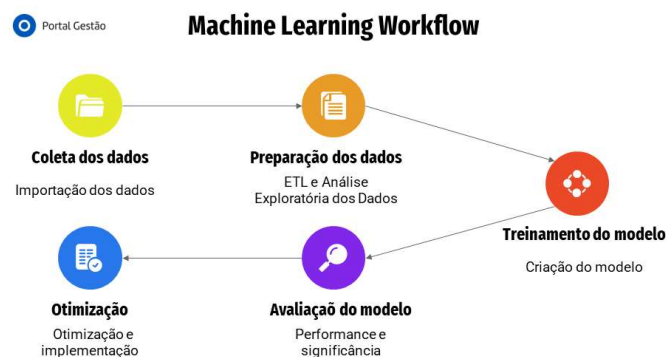


Figura 7: Fluxo de processamento de ML. Fonte [44]

3.1.1 Preparação e Transformação dos dados

Na análise exploratória dos dados, foram encontrados alguns campos vazios nas colunas que representam a identificação do usuário (*id*), cartão (pictograma) e localização. Essas linhas foram removidas do *dataset*, pois essas informações são necessárias para o treinamento dos modelos. Foram extraídas 267.403 linhas em que os usuários eram nulos, 291.170 linhas sem a escolha final do usuário e 340.443 linhas onde a localização era nula. Após essa primeira filtragem, o conjunto de dados passou a conter 465.254 linhas, com 281 usuários únicos.

As colunas relevantes para a pesquisa, são:

- *user_uuid*, representando a identificação do usuário, sendo utilizada para o treinamento personalizado;
- *click_location*, contém a localização onde o usuário interagiu com o aplicativo, representada por latitude e longitude, nos formatos -01.0101, -45.4545;
- *timestamp* contém data, dia e hora da interação, armazenada no formato 1736865929382000;
- *card_written_text*, se refere ao pictograma escolhido pelo usuário, contendo frases textuais, exemplo: “comer”, “sair”. Os dados dessa coluna foram convertidos em letras minúsculas para evitar que houvesse diferença entre palavras idênticas.

Foram criadas colunas a partir das já existentes:

- *week_day*, indica o dia da semana (*Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday*), gerado a partir da coluna *timestamp*. Essa coluna foi utilizada como contexto no treinamento;
- *hour*, representa as horas do dia (de 0 a 23). Não houve necessidade de ajustes para diferentes fusos. Informação extraída da coluna *timestamp*;
- *year_num*, com o ano, formato 2025. Utilizado para a separação dos dados por número de semana. Informação gerada a partir da coluna *timestamp*;
- *week_num*, representa o número da semana (1 a 52), também utilizado na separação dos dados por semana. Informação gerada a partir da coluna *timestamp*;

- *period_day*, com o período do dia em que aconteceu a interação, sendo gerada a partir da coluna *hour*. Os períodos foram definidos em intervalos de três horas, exceto a madrugada (*midnight*). Os períodos ficaram representados por *dawn* (6h-8h), *morning* (9h-11h), *noon* (12h-14h), *afternoon* (15h-17h), *evening* (18h-20h), *night* (21h-23h), *midnight* (0h-5h). Esses horários foram mantidos no fuso local, sem necessidades de conversão. Essa coluna foi utilizada como contexto no treinamento;
- *cluster*, essa coluna foi criada a partir da clusterização com ajuda do algoritmo OPTICS, um método baseado em densidade que ordena os pontos de um conjunto de dados para revelar estruturas de agrupamento [45]. Essa coluna foi utilizada como contexto no treinamento;
- *week_order*, com a ordem das semanas. Calculada a partir das colunas *week_num* e *year_num*, atribuindo valores em ordem crescente, onde semanas mais antigas recebem números menores.

A adaptação da coluna *period_day* foi importante para proporcionar um contexto mais adequado ao momento do dia em que as escolhas foram feitas. Considerar apenas as horas tornaria a quantidade de contextos muito difusa e difícil de adaptar.

3.1.2 Filtragem e Visualização dos Dados

Com o objetivo de obter um conjunto de dados diversificado, foram definidos critérios para evitar viés nos resultados ou comprometer as métricas de desempenho.

Os critérios considerados para filtragem dos dados (por usuário):

- Mínimo de cinquenta cliques: Uma barreira geral para ser considerado na pesquisa, um uso menor que esse, indica que o usuário não usou a aplicação por tempo suficiente. Com esse filtro foram removidas 141.311 linhas do *dataset*.
- Dez cliques em locais distintos: A variedade de locais permite que o modelo aprenda melhor com os diferentes contextos. Com esse filtro foram removidas do *dataset* 11.106 linhas.
- Vinte cliques em horários distintos: Assim como no critério anterior, é necessário que haja interações em diferentes horários a fim de capturar

o comportamento do usuário durante seu dia. Com esse filtro foram removidas do *dataset* 10.615 linhas.

Duas últimas filtragens foram realizadas, excluindo do *dataset* os usuários que possuíam menos mil linhas, reduzindo do *dataset* 4.521 linhas e 11 usuários. Também foram removidos os usuários que possuíam menos de três semanas de utilização do aplicativo, esse filtro resultou na exclusão de 1 usuário e 3.473 linhas.

Essas exclusões, foram realizadas para garantir que os usuários selecionados tivessem dados suficientes para o treinamento e teste, visto que esse processo foi feito de forma personalizada por usuário.

O conjunto de dados passou a conter 294.229 linhas, com um total de 32 usuários únicos. A figura 8 apresenta a quantidade de cartões por usuário, enquanto a figura 9 exibe a quantidade de localizações únicas por usuário.

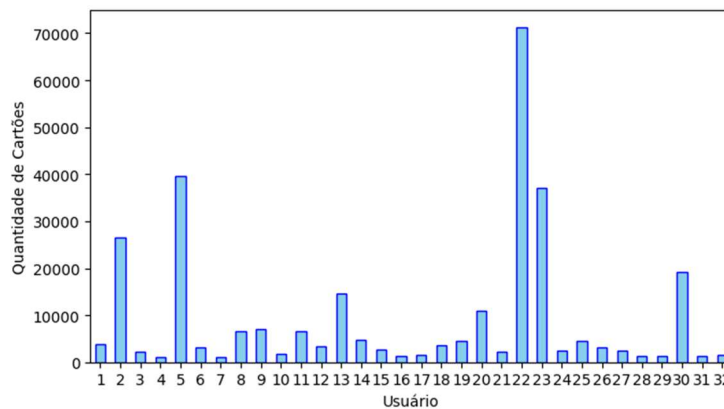


Figura 8: Distribuição da quantidade de cartões por usuário

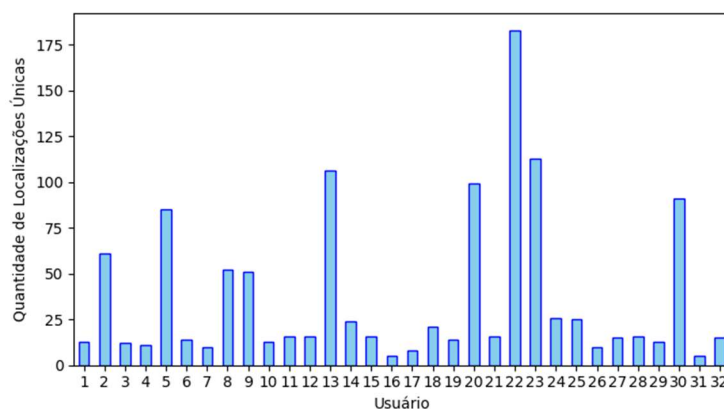


Figura 9: Distribuição da quantidade de localizações distintas por usuário

3.1.3 Separação dos dados

O método de separação dos dados ocorreu por meio de validação cruzada, na qual o conjunto de dados foi dividido em subconjuntos de treino e teste [46]. No conjunto de treino, foram consideradas as características de contexto e os cartões (*labels*), enquanto os dados de teste também foram separados em contexto e cartão.

O conjunto passou pelo processo de treinamento do modelo, onde os parâmetros foram ajustados para minimizar o erro e melhorar o resultado [46]. Na última etapa, o modelo treinado foi utilizado para prever os dados de teste, em seguida, validado por métricas que avaliam o desempenho. A figura 7 apresenta uma representação do processo de separação.

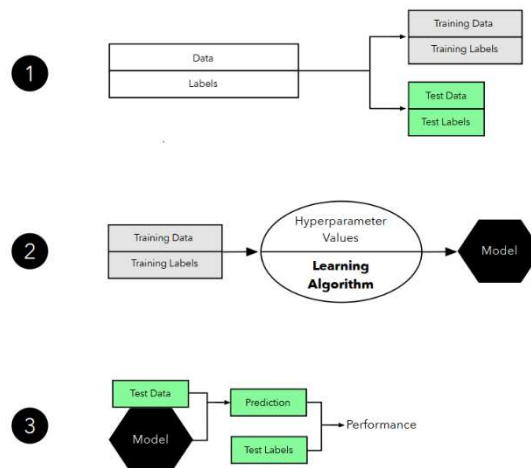


Figura 10: Exemplo de separação dos dados na validação cruzada. Fonte: [46]

O objetivo do estudo é avaliar os modelos de forma personalizada. Para isso, o treinamento e o teste foram realizados por usuário. Cada um dos 32 usuários finais, foram separados por semanas. Para os dados de teste foi considerada as duas semanas mais recentes em que o aplicativo foi utilizado e para treino as semanas anteriores. Essas informações foram obtidas pela coluna *week_order*. O objetivo é que os modelos aprendam com os dados das semanas anteriores e gerem recomendações para um período futuro.

Devido a essas condições, a proporção entre treino e teste dos usuários não seguiu as porcentagens normalmente utilizadas em validações cruzadas (80/70% para treino e 20/30% para teste) [46]. A figura 11 apresenta as proporções de treinamento e teste de cada usuário.

ID	Treino	% Treino	Teste	% Teste	ID	Treino	% Treino	Teste	% Teste	ID	Treino	% Treino	Teste	% Teste
22	69396	97%	1960	3%	6	2626	85%	469	15%	15	1620	62%	984	38%
8	6317	96%	231	4%	12	2881	83%	580	17%	32	846	56%	664	44%
20	10579	96%	389	4%	28	976	81%	228	19%	16	735	54%	620	46%
5	37865	96%	1660	4%	26	2414	77%	714	23%	17	824	51%	784	49%
14	4484	95%	242	5%	24	1766	75%	586	25%	11	3014	46%	3550	54%
9	6667	95%	371	5%	7	755	74%	262	26%	31	130	10%	1208	90%
18	3459	95%	198	5%	29	929	73%	341	27%					
3	2082	94%	132	6%	27	1801	72%	686	28%					
13	13754	94%	881	6%	1	2721	72%	1057	28%					
23	34500	93%	2639	7%										
21	2121	92%	174	8%										
25	4241	92%	364	8%										
10	1532	91%	158	9%										
2	23679	89%	2811	11%										
4	900	89%	110	11%										
30	16841	88%	2264	12%										
19	3872	87%	585	13%										

Figura 11: Proporção entre treinamento e teste por usuário

3.2 Modelos

Os modelos foram construídos utilizando bibliotecas como o TF, para o RNA e *Sklearn* [48] para transformações de variáveis. Ambas as bibliotecas são amplamente utilizadas em aprendizado de máquina e oferecem ferramentas robustas nesse sentido. Como contexto, foram utilizadas as colunas *period_day*, *week_day*, *cluster* enquanto a coluna *card* como *label*.

3.2.1 Naïve Bayes

O modelo foi construído utilizando o MultinomialNB [46], um algoritmo utilizado em problemas de classificação em que os dados são representados como contagens ou frequência de eventos.

O primeiro passo do modelo consistiu na codificação das características categóricas. Para transformar as informações contextuais em representações numéricas, foram utilizadas instâncias da classe *LabelEncoder*, nas variáveis de contexto e *label* (global). Após essas transformações os dados passam a ser representados pelas variáveis *week_day_enc*, *period_day_enc*, *cluster_enc* e *card_enc*.

Cada valor categórico foi convertido para um índice inteiro único, garantindo que os dados os dados pudessem ser utilizados como entrada no modelo que opera sobre valores numéricos.

Dentro de uma estrutura de repetição, o modelo treinou para cada usuário, durante esse processo, os cartões do usuário foram reindexados em um intervalo

menor por um *LabelEncoder* (local) representado pela variável *card_local_enc*, considerando apenas os cartões que aquele usuário viu. Caso um cartão presente no conjunto de teste não existisse nos dados de treino, ele era removido para evitar erros ao alimentar o modelo. Além disso o treinamento utilizou um parâmetro alpha 0.5, que regula a suavização de *Laplace* para evitar probabilidades nulas. Após o treinamento foi realizada a avaliação do modelo.

3.2.2 Rede Neural Artificial

O modelo foi construído utilizando a arquitetura de duas torres, sendo treinado individualmente para cada usuário. Inicialmente, as informações categóricas foram transformadas em representações numéricas, o *LabelEncoder* foi utilizado para a coluna *card*, representado pela variável *card_enc*. A partir dessa variável, foi gerada a lista de cartões únicos (*num_cards*). As variáveis *period_day*, *week_day* e *cluster* foram transformadas com o *one-hot encoding*, formando o conjunto de características *context_features*, contendo as variáveis de contexto.

A primeira torre foi responsável por processar as características de contexto, por meio de uma camada densa com 32 neurônios e ativação ReLU e uma segunda camada densa com 16 neurônios de mesma ativação. A segunda torre processou as informações dos cartões através de uma camada de incorporação com 16 dimensões e uma camada *flatten* que converte a saída da incorporação em um vetor unidimensional.

Em seguida as duas saídas foram concatenadas passando por duas novas camadas densa com 32 e 16 neurônios e mesma ativação. Na última camada os *num_cards* são utilizados pela ativação *softmax*, para gerar a distribuição de probabilidades dos cartões recomendados. O modelo utilizou otimizador Nadam, a função de perda *sparse_categorical_crossentropy*, e *TopKCategoricalAccuracy* como métrica.

3.2.3 Aprendizagem Federada

Na abordagem federada, os modelos locais são treinados separadamente e depois agregados em um modelo global. Inicialmente o modelo global foi criado seguindo as mesmas etapas que a arquitetura duas torres, incluindo a transformação

das variáveis categóricas e a passagem pelas camadas densas de ambas as torres para a criação do modelo global.

Após a criação desse modelo, foram treinados os modelos locais para cada usuário por um total de dez épocas. Em seguida os pesos de cada modelo foram armazenados e agregados por meio da função *FedAvg*. Como visto anteriormente essa técnica calcula a média dos pesos de cada camada e os utiliza para atualizar os pesos do modelo global, garantindo que as informações nas bases locais incorporem de forma equilibrada.

Depois que as médias dos pesos são calculadas e aplicados ao modelo global, é realizado um ajuste fino para o modelo híbrido. Nessa fase, cada usuário recebeu uma versão do modelo global atualizado e realizou um novo treinamento com seus próprios dados. A ideia para esse modelo era permitir o modelo aprender características de cada usuário sem comprometer a generalização obtida do modelo global. Esse processo foi repedido por três rodadas.

3.2.4 Parâmetros e Métricas De Avaliação

Os modelos foram testados com os parâmetros descritos na tabela 1, sendo destacados aqueles que apresentaram os melhores resultados. Para avaliação, foi utilizada a abordagem de *holdout* como forma de avaliação dos modelos de AM.

Modelo	Parâmetros	Valores
NB	Alpha Modelo	0.01 , 0.1, 0.5, 1.0, 10.0 MultinomialNB
RNA	Otimizador Camadas Neurônios Perda Métrica Treino	Adam, Nadam , SGD, RMSprop 4 16 32 Sparse_categorical_crossentropy Sccuracy, sparseCategoricalAcc , TopKCategoryalAccuracy(K)
AF	Otimizador Camadas Neurônios Perda Métrica Treino Época local Rodadas Época ajuste fino	Adam, Nadam , SGD, RMSprop 4 16, 32 Sparse_categorical_crossentropy Accuracy, sparseCategoricalAcc , TopKCategoryalAccuracy(K), 5, 10 3 1

Tabela 1: Modelos e valores dos parâmetros utilizados.

O *recall* foi a métrica utilizada para validação dos modelos, aplicados na forma de *recall@k*. Essa métrica avalia a precisão das predições considerando os K itens mais prováveis e os compara com as reais K escolhas [48]. Essa métrica é adequada para esse contexto, por não ser necessariamente importante sugerir apenas um único item. Em ambientes reais os usuários gostariam de ter mais opções de sugestões, logo comparar os K primeiros itens em relação aos que foram escolhidos se torna uma boa opção.

$$Recall@K = \frac{\text{Número de itens relevantes em } K}{\text{Total de itens em } K} \quad (2)$$

Em todos os modelos foi utilizada a mesma métrica, analisando cada usuário ao comparar os K itens recomendados com os K escolhidos de fato, foram aplicadas as variações de *recall@1*, comparação feita entre o primeiro item recomendado e o primeiro item do usuário *recall@5*, segue a mesma lógica, mas para cinco itens e *recall@10* para dez itens.

Para as classes reais, foi considerado a quantidade de utilizações como forma de ranqueamento, enquanto nos modelos foram consideradas as probabilidades atribuídas a cada item, da maior para a menor. No modelo de aprendizagem federada o *recall@k* foi utilizado para calcular os K primeiros itens, considerando os modos: global, local e híbrido.

4 EXPERIMENTOS E RESULTADOS

Como visto anteriormente, os dados utilizados no treinamento não seguem proporções padronizadas. Para contornar essa situação e possibilitar comparações em condições homogêneas, os resultados foram agrupados nessas três faixas. Faixa menor para usuários em que a distribuição dos testes foi menor que 15%, enquanto a faixa intermediária ficou entre 15% e 35%, a faixa mais alta acima de 35%.

4.1 Resultados

O modelo NB apresentou resultados baixos, possivelmente influenciadas pelas distorções na distribuição dos dados entre usuários e semanas. Essa abordagem obteve o menor desempenho entre os modelos testados, o que já era

esperado, já que o modelo teve dificuldade em lidar com os dados não vistos durante o treinamento, impactando nos resultados observados na figura 12. Em geral, o modelo não teve grandes variações entre as faixas, desempenhando melhor na faixa média, com distribuição mais próxima do padrão, chegando à média 0,28 para *recall@10*.

O modelo conseguiu recomendar cartões personalizados aos usuários de maneira satisfatória em ocasiões de um bom *recall*, caso do usuário 1 que teve *recall@10* igual 0,75.

FAIXA	MÉDIA	DESVIO	RECALL	MODELO
BAIXA	0,06	0,07	1	NB
	0,19	0,12	5	
	0,27	0,13	10	
MÉDIA	0,06	0,05	1	
	0,18	0,15	5	
	0,28	0,22	10	
ALTA	0,08	0,11	1	
	0,16	0,10	5	
	0,23	0,09	10	

Figura 12: Resultado do modelo *Naïve Bayes*

Utilizando as variáveis de contexto, *cluster* 16, o início da tarde como período do dia e na terça-feira o modelo conseguiu prever boa parte dos cartões reais do usuário 1 na tabela 2.

Sugestões do modelo NB <i>call@10</i>	Escolhas reais
'comer', 'guloseimas ', 'pedro', 'fandangos ', 'mauricio', 'atividades ', 'confetes ', 'cores', 'frutas', 'cheetos'	'fandangos ', 'guloseimas ', 'comer', 'pedro', 'mauricio', 'atividades ', 'fome', 'banheiro', 'frutas', 'murilo'

Tabela 2: Recomendação do modelo NB

No modelo de rede neural os resultados se mostram muito melhores, com a faixa mais baixa se destacando, isso foi reflexo da alta quantidade de dados disponíveis para treino, com todas as médias de *recall* performando bem embora também tenha atingido altos índices de desvio padrão. As faixas médias e alta vistas na figura 13, tiveram resultados moderados a baixos, considerando seu maior *recall* de 0,49 na faixa média. Individualmente o usuário 1 com a melhor proporção na

distribuição obteve taxas tão altas quanto os modelos com muito mais dados para treino.

O modelo conseguiu alcançar um de recomendação superior ao NB, de forma relevante ao usuário, considerando o mesmo usuário e contextos visto no modelo anterior, conseguiu prever uma taxa superior (maior que 0,9) e mesmo casos de recomendações incorretas para o contexto, fez sugestões que estavam dentro da lista de cartões do usuário, gerando uma sugestão aproximada.

FAIXA	MÉDIA	DESVIO	RECALL	MODELO
BAIXA	0,69	0,37	1	RNA
	0,73	0,34	5	
	0,76	0,31	10	
MÉDIA	0,25	0,27	1	
	0,41	0,27	5	
	0,49	0,27	10	
ALTA	0,12	0,14	1	
	0,27	0,23	5	
	0,35	0,23	10	

Figura 13: Resultado do modelo RNA

Com a abordagem federada, o *recall* foi calculado para os modelos globais, locais e híbrido, permitindo uma comparação direta entre recomendações generalistas e personalizadas. O modelo global teve o pior desempenho entre as três abordagens, com um *recall@10* de 0,23 na menor faixa. Já o modelo local conseguiu 0,89 no *recall@10* da menor faixa, com o segundo menor desvio padrão, demonstrando como essa faixa obteve bons resultados individuais.

O modelo híbrido teve um resultado mediano, com *recalls* variando entre 0,28 e 0,42 na faixa média, esse resultado mostra que considerar as informações dos outros usuários pode corromper a capacidade do modelo em realizar recomendações relevantes. Mesmo com esse desempenho um pouco menor, ainda foi muito superior ao modelo global. Reforçando a importância de se ter modelos personalizados.

Em geral o modelo de RNA obteve os melhores resultados em comparação aos demais, considerando tanto o *recall* médio quanto as menores taxas de desvio padrão. Indicando uma melhor consistência nos resultados.

5 CONCLUSÃO

O presente estudo teve como objetivo agregar conhecimento no campo do aprendizado de máquina, tendo como foco as aplicações de comunicação aumentativa alternativa, representadas pelo aplicativo Livox. O intuito principal foi aprimorar sistemas de recomendação sensíveis ao contexto, dada sua importância para pessoas que necessitam de ferramentas modernas e práticas para auxiliar em suas comunicações e melhorar seu convívio social.

O estudo realizado em [10] onde modelos de redes neurais foram criados com uma abordagem global, mas sem bons resultados, abriu a possibilidade de se aprofundar nesse tema e verificar como um procedimento similar, mas focado em recomendações personalizadas se comportaria. Os resultados obtidos foram positivos, contribuindo para que mais estudos possam ser realizados nesse sentido, promovendo um ambiente favorável para aplicações especializadas não só no auxílio da comunicação, mas também nas necessidades que essas pessoas podem ter.

A utilização de diferentes modelos demonstrou que a aplicação do aprendizado de máquina em sistemas de recomendação, são amplas, e que modelos com diferentes abordagens podem gerar resultados distintos, ainda que sejam utilizados os mesmos tratamentos nos dados.

Para alcançar esses resultados, foram realizados estudos sobre os diferentes modelos e abordagens, seus princípios básicos de funcionamento e as ferramentas auxiliaram nesse processo. Foram conduzidos treinamentos e testes com diferentes parâmetros, permitindo comprovar que modelos personalizados apresentam melhor desempenho que modelos globais, considerando os ambientes de SRSC e CAA.

5.1 Limitações

Embora o modelo de recomendação sensível ao contexto focado no usuário, possa permitir recomendações personalizadas, foi possível notar que a escassez de dados representa um desafio. Esse problema não se limita a quantidade, mas também em qualidade das informações disponíveis. Dependendo exclusivamente dos dados de cada usuário pode dificultar a geração de recomendações personalizadas a novos usuários ou situações em que há poucas interações registradas.

Entretanto mesmo com uma base reduzida de dados pré-processados, alguns casos obtiveram um resultado satisfatório. Esse aspecto sugere que o modelo pode ter maior adaptabilidade as mudanças no comportamento do usuário, sem depender de dados macros, que nem sempre são relevantes. Comportamento observado nos modelos AF.

Neste estudo não foi incluído modelo global para a abordagem RNA por já existir trabalho que aborda esse tema [10], embora pudesse ser interessante verificar nos modelos de rede neural e *Naïve Bayes* o comportamento dos modelos globais em relação aos dados trabalhados, assim como foi possível ver no modelo de aprendizagem federada.

5.2 Trabalhos Futuros

O tratamento dos dados é primordial em aplicações de AM, ter boas informações podem levar a bons resultados. Nessa linha, utilizar modelos de linguagem de larga escala (*Large Language Models*, LLM) pode ter um impacto muito positivo, alguns casos encontrados durante a exploração de dados como palavras sem muito sentido, “sim”, “quero”, “quero...” se repetem por diversas vezes e esses modelos seriam capazes de categorizar essas classes de uma forma melhor, permitindo que palavras realmente válidas façam parte do modelo.

Outro ponto interessante de ser estudado seria as aplicações de mais de um modelo, visto um pouco no model AF, mais com bastante potencial a ser explorado. Utilizar modelos que interajam com o usuário em tempo real ajustando a recomendação, muito utilizados na aprendizagem por reforço, podem ser um bom caminho.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Tomasello, Michael. *Origins of human communication*. MIT press, 2010.
- [2] Beukelman, David R., and Pat Mirenda. *Augmentative and alternative communication*. Baltimore: Paul H. Brookes, 1998.
- [3] Millar, Diane C., Janice C. Light, and Ralf W. Schlosser. "The impact of augmentative and alternative communication intervention on the speech production of individuals with developmental disabilities: A research review." (2006).
- [4] Elsahar, Yasmin, et al. "Augmentative and alternative communication (AAC) advances: A review of configurations for individuals with a speech disability." *Sensors* 19.8 (2019): 1911.
- [5] Ko, Hyeyoung, Suyeon Lee, Yoonseo Park, and Anna Choi. "A survey of recommendation systems: recommendation models, techniques, and application fields." *Electronics* 11, no. 1 (2022): 141.
- [6] Cheng, Heng-Tze, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishikesh Aradhye, Glen Anderson et al. "Wide & deep learning for recommender systems." In *Proceedings of the 1st workshop on deep learning for recommender systems*, pp. 7-10. 2016.
- [7] Bell, Robert, Yehuda Koren, and Chris Volinsky. "Modeling relationships at multiple scales to improve accuracy of large recommender systems." In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 95-104. 2007.
- [8] Verbert, Katrien, Nikos Manouselis, Xavier Ochoa, Martin Wolpers, Hendrik Drachsler, Ivana Bosnic, and Erik Duval. "Context-aware recommender systems for learning: a survey and future challenges." *IEEE transactions on learning technologies* 5, no. 4 (2012): 318-335.
- [9] Livox (2024), <https://livox.com.br/br/>, acessado em 16-Julho-2024.
- [10] Silva, Ulisses C. "Recomendação Sensível ao Contexto para Comunicação Aumentativa e Alternativa Baseada em Aprendizagem de Máquina." UFRPE, 2024
- [11] Light, Janice, David McNaughton, David Beukelman, Susan Koch Fager, Melanie Fried-Oken, Thomas Jakobs, and Erik Jakobs. "Challenges and opportunities in augmentative and alternative communication: Research and technology development to enhance communication and participation for individuals with complex communication needs." *Augmentative and Alternative Communication* 35, no. 1 (2019): 1-12.
- [12] Dynavox, T.: Tobiidynavox (2024), <https://www.tobiidynavox.com>, acessado em 16-Julho-2024.
- [13] Democracy, E.: Eyegaze (2024), <https://eyegaze.com>, acessado em 16-Julho-2024.
- [14] Ware, A.: Proloquo2go (2024), <https://www.assistiveware.com>, acessado em 16-Julho-2024.

- [15] Neamtu, Rodica, André Camara, Carlos Pereira, and Rafael Ferreira. "Using artificial intelligence for augmentative alternative communication for children with disabilities." In *Human-Computer Interaction—INTERACT 2019: 17th IFIP TC 13 International Conference, Paphos, Cyprus, September 2–6, 2019, Proceedings, Part I* 17, pp. 234-243. Springer International Publishing, 2019.
- [16] Javed, Umair, Kamran Shaukat, Ibrahim A. Hameed, Farhat Iqbal, Talha Mahboob Alam, and Suhuai Luo. "A review of content-based and context-based recommendation systems." *International Journal of Emerging Technologies in Learning (iJET)* 16, no. 3 (2021): 274-306.
- [17] Schein, Andrew I., Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. "Methods and metrics for cold-start recommendations." In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 253-260. 2002.
- [18] Abbas, Assad, Limin Zhang, and Samee U. Khan. "A survey on context-aware recommender systems based on computational intelligence techniques." *Computing* 97 (2015): 667-690.
- [19] Adomavicius, Gediminas, and Alexander Tuzhilin. "Context-aware recommender systems." In *Recommender systems handbook*, pp. 217-253. Boston, MA: Springer US, 2010.
- [20] Dey, Anind K. "Understanding and using context." *Personal and ubiquitous computing* 5 (2001): 4-7.
- [21] Haruna, Khalid, Maizatul Akmar Ismail, Suhendroyono Suhendroyono, Damiasih Damiasih, Adi Cilik Pierewan, Haruna Chiroma, and Tutut Herawan. "Context-aware recommender system: A review of recent developmental process and future research direction." *Applied Sciences* 7, no. 12 (2017): 1211.
- [22] Isinkaye, Folasade Olubusola, Yetunde O. Folajimi, and Bolande Adefowoke Ojokoh. "Recommendation systems: Principles, methods and evaluation." *Egyptian informatics journal* 16, no. 3 (2015): 261-273.
- [23] Rish, Irina. "An empirical study of the naive Bayes classifier." In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, no. 22, pp. 41-46. 2001.
- [24] Zhang, Harry. "The optimality of naive Bayes." *Aa* 1, no. 2 (2004): 3.
- [25] Ghazanfar, Mustansar, and Adam Prugel-Bennett. "An improved switching hybrid recommender system using naive bayes classifier and collaborative filtering." (2010).
- [26] Gurney, Kevin. *An introduction to neural networks*. CRC press, 2018.
- [27] Bishop, Christopher M., and Nasser M. Nasrabadi. *Pattern recognition and machine learning*. Vol. 4, no. 4. New York: springer, 2006.
- [28] AIML (2024), <https://aiml.com/what-is-the-basic-architecture-of-an-artificial-neural-network-ann>, acessado em 19-Julho-2024.
- [29] Ma, Jiaqi, Zhe Zhao, Xinyang Yi, Ji Yang, Minmin Chen, Jiaxi Tang, Lichan Hong, and Ed H. Chi. "Off-policy learning in two-stage recommender systems." In *Proceedings of The Web Conference 2020*, pp. 463-473. 2020.

- [30] Covington, Paul, Jay Adams, and Emre Sargin. "Deep neural networks for youtube recommendations." In *Proceedings of the 10th ACM conference on recommender systems*, pp. 191-198. 2016.
- [31] Rish, Irina. "An empirical study of the naive Bayes classifier." In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, no. 22, pp. 41-46. 2001.
- [32] Zhang, Libo, Tiejian Luo, Fei Zhang, and Yanjun Wu. "A recommendation model based on deep neural network." *IEEE Access* 6 (2018): 9454-9463.
- [33] Li, Qinbin, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. "A survey on federated learning systems: Vision, hype and reality for data privacy and protection." *IEEE Transactions on Knowledge and Data Engineering* 35, no. 4 (2021): 3347-3366.
- [34] Li, Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. "A review of applications in federated learning." *Computers & Industrial Engineering* 149 (2020): 106854.
- [35] Muhammad, Khalil, Qinqin Wang, Diarmuid O'Reilly-Morgan, Elias Tragos, Barry Smyth, Neil Hurley, James Geraci, and Aonghus Lawlor. "Fedfast: Going beyond average for faster training of federated recommender systems." In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1234-1242. 2020.
- [36] Rieke, Nicola, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R. Roth, Shadi Albarqouni, Spyridon Bakas et al. *The future of digital health with federated learning*. *NPJ Digital Medicine*, 3, 119. 2020.
- [37] Naumov, Maxim, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang et al. "Deep learning recommendation model for personalization and recommendation systems." *arXiv preprint arXiv:1906.00091* (2019).
- [38] Pereira, Jayr A., Sheyla de Medeiros, Cleber Zanchettin, and Robson do N. Fidalgo. "Pictogram prediction in alternative communication boards: a mapping study." In *Simpósio Brasileiro de Informática na Educação (SBIE)*, pp. 705-717. SBC, 2022.
- [39] Evangeline, S. Betsybha, and Anitha Dhakshina Moorthy. "Improving Communication for People with Speech Disabilities: Exploring Context-Aware Recommendation Systems in Assistive Technology." In *2024 International Conference on Intelligent Computing and Emerging Communication Technologies (ICEC)*, pp. 1-6. IEEE, 2024.
- [40] TensorFlow (2025), <https://www.tensorflow.org/?hl=pt-br>, acessado em 20-Fevereiro-2025.
- [41] Moresi, Eduardo. "Metodologia da pesquisa." *Brasília: Universidade Católica de Brasília* 108, no. 24 (2003): 5.
- [42] Lei^o 13.709, https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm, acessado em 24-Fevereiro-2025.
- [43] García, Salvador, Julián Luengo, and Francisco Herrera. *Data preprocessing in data mining*. Vol. 72. Cham, Switzerland: Springer International Publishing, 2015.

- [44] LinkedIn (2025), <https://pt.linkedin.com/pulse/modelo-preditivo-azure-machine-learning-demanda-por-lupchinski>, acessado em 04-Março-2025.
- [45] Ankerst, Mihael, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. "OPTICS: Ordering points to identify the clustering structure." *ACM Sigmod record* 28, no. 2 (1999): 49-60.
- [46] Raschka, Sebastian. "Model evaluation, model selection, and algorithm selection in machine learning." *arXiv preprint arXiv:1811.12808* (2018).
- [47] Sklearn (2025), https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html, acessado em 07-Março-2025.
- [48] Buckland, Michael, and Fredric Gey. "The relationship between recall and precision." *Journal of the American society for information science* 45, no. 1 (1994): 12-19.