

DESENVOLVIMENTO DE UM SISTEMA DE RECOMENDAÇÃO DE ORIENTADORES PARA TCCS UTILIZANDO UM MODELO PREDITIVO BASEADO EM NAIVE BAYES

Mairon Santana do Nascimento*

Ruan dos Santos Gondim**

Raimundo Valter Costa Filho***

RESUMO

A educação desempenha um papel importante nas economias globais, sendo vital para o desenvolvimento das potências mundiais. No entanto, o mau funcionamento dos sistemas educacionais pode restringir o acesso ao conhecimento e prejudicar a inovação tecnológica. A alta taxa de evasão escolar é prejudicial à pesquisa tecnológica do país. Estudos realizados no IFSP apontam que os alunos tiveram dificuldades durante a elaboração do Trabalho de Conclusão de Curso (TCC), sendo esse um dos principais motivos que os impediram de concluir o curso. Outra pesquisa destaca a complexidade dessa etapa, com observação de estresse e, em alguns casos, crises de ansiedade nos alunos. Os estudantes quando entrevistados, informaram que ter um orientador com domínio na área de pesquisa do trabalho é um facilitador nessa fase. Diante desse cenário, o presente estudo propôs a implementação de um sistema de recomendação entre alunos e professores orientadores, trazendo uma recomendação a partir de temas em comum, a fim de melhorar a experiência dos alunos na escolha de orientadores e na conclusão do TCC. Para atingir este objetivo, realizou-se um trabalho de análise de dados sobre o ecossistema de TCCs defendidos do IFCE - campus Aracati. Além disso, foram conduzidos estudos e adaptações com o algoritmo *Naive Bayes*, a fim de montar um modelo preditivo capaz de recomendar os orientadores ideais para os alunos. Como um resultado adicional, foi desenvolvido também uma API capaz de gerenciar as recomendações e oferecer ferramentas de apoio para a recomendação feita pela IA.

Palavras-chave: Inteligência Artificial. Sistemas de Recomendação. Análise de Dados. Trabalho de Conclusão de Curso.

* Autor, Graduando em Bacharelado em Ciência da Computação, Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE), Aracati, Ceará, Brasil. E-mail: maironnscmt@gmail.com

** Orientador, Mestre em Ciência da Computação pela Universidade Federal Rural do Semi-Árido - URFESA, Docente do Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE), Aracati, Ceará, Brasil. rugondim@gmail.com

*** Coorientador, Doutor em Engenharia de Teleinformática pela Universidade Federal do Ceará - UFC, Docente do Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE), Aracati, Ceará, Brasil. valter.costa@ifce.edu.br

ABSTRACT

Education plays an important role in global economies, being vital for the development of world powers. However, malfunctioning educational systems can restrict access to knowledge and hinder technological innovation. The high rate of school dropout is detrimental to the country's technological research. Studies conducted at IFSP indicate that students faced difficulties during the elaboration of their Final Paper, being this one of the main reasons preventing them from completing the course. Another research emphasizes the complexity of this stage, noting stress and, in some cases, anxiety crises in students. When interviewed, students reported that having an advisor with expertise in the research area of the paper is a facilitator in this phase. In light of this scenario, the present study proposed the implementation of a recommendation system between students and advising professors, providing recommendations based on common themes to enhance students' experience in choosing and completing their Final Paper. To achieve this goal, a data analysis was conducted on the ecosystem of Final Papers at IFCE - Aracati campus. Additionally, studies and adaptations were made with the *Naive Bayes* algorithm to create a predictive model capable of recommending the ideal advisors for students. As an additional outcome, an API was also developed to manage recommendations and provide support tools for the recommendations made by the AI.

Keywords: Artificial Intelligence. Recommendation Systems. Data analysis.

1 INTRODUÇÃO

A educação desempenha um papel significativo nas atividades econômicas globais, devido a mobilização de recursos físicos e econômicos, além de diversas pessoas que colaboram e são necessárias para seu funcionamento (CANZIANI et al., 2015). A educação também é um aspecto fundamental para o desenvolvimento das grandes potências mundiais (DANA, 2017). O mau funcionamento dos sistemas de ensino geram restrições de acesso ao conhecimento e, portanto, diminui a possibilidade de gerar algum tipo de inovação tecnológica (COOKE; URANGA; ETXEBARRIA, 1997). Assim, a alta taxa de evasão escolar prejudica consideravelmente a pesquisa tecnológica de um país, principalmente no Brasil, um país emergente que importa bastante tecnologia do exterior.

No contexto do ensino superior, o IFCE possui uma alta taxa de evasão. São mais de 24 mil evadidos desde 2013. No campus Aracati, os números são igualmente preocupantes, no total são cerca de 2600 evasões entre cursos técnicos superiores, o equivalente a 45,84% (IFCE, 2023). (FAUSTINO-FERBER, 2021) realizou uma pesquisa com alunos evadidos do campus de São Paulo do IFSP, onde 42% dos ex-alunos concluíram os créditos nas disciplinas, porém não finalizaram o curso, ou seja, desistiram durante a etapa Trabalho de Conclusão de Curso (TCC).

O TCC é exigido por diversos cursos como um estágio fundamental para o processo de formação do estudante, visando prepará-lo para o mercado de trabalho (GUIMARÃES; SOBRINHO, 2020). Para muitos alunos, esta é uma etapa acadêmica muito atribulada e marcada por sentimentos negativos. Segundo (JUNIOR et al., 2022) em sua pesquisa sobre emoções e sentimentos dos alunos neste estágio da vida acadêmica, nota-se que as 20 palavras mais citadas pelos alunos de TCC representam sentimentos negativos, tais como frustração, medo, raiva, preguiça e entre outros.

Na pesquisa de (FAUSTINO-FERBER, 2021), dentre as causas relacionadas à desistência do aluno, foram citados como um dos principais motivos os didáticos-pedagógicos, muitas vezes relacionados a aspectos da orientação do TCC. No mesmo estudo, alunos citam que orientadores não ofereceram o suporte necessário para concluir o projeto, dificultando assim a finalização do trabalho.

De acordo com (GUIMARÃES; SOBRINHO, 2020) em seu estudo sobre o processo de construção do TCC, um dos principais fatores que dificultam o desenvolvimento do projeto é a procura por um orientador. Por outro lado, ter um orientador definido e que possui domínio e *expertise* sobre o tema é um grande facilitador deste processo.

Hoje, com o surgimento do conceito de *Big Data*, surge com ele também os Sistemas de Recomendação, que consistem na utilização dos dados de interação do usuário para filtrar conteúdo. Essa solução é aplicada em diversos contextos, entre eles o comércio eletrônico, serviços de *streaming* sob demanda, redes sociais (TAKAHASHI; JR, 2015) e, inclusive, no contexto da educação. Uma das diversas aplicações dos sistemas de recomendação tem sido a filtragem de conteúdos didáticos para estudantes, trazendo um conteúdo totalmente personalizado às necessidades do aluno (ZHU, 2023). As tendências do momento apontam para um futuro onde os ecossistemas educacionais incluem cada vez mais tecnologias da IA (Inteligência Artificial) e suas vertentes para as mais diversas aplicações (VICARI, 2021).

Nesse contexto, a proposta deste trabalho foi desenvolver um sistema de recomendação entre alunos e professores orientadores. O algoritmo trabalhou a partir de uma base de dados histórica dos alunos concludentes, com seus temas e orientadores. Além disso, foi feito um estudo sobre o impacto na recomendação quando incluído o tempo de experiência de cada professor na matéria. O sistema foi projetado utilizando os dados do curso de Bacharelado de Ciência da Computação do IFCE - Campus Aracati, entretanto o mesmo poderá ser adaptado e eventualmente utilizado para qualquer curso, desde que tenha os dados históricos dos TCCs defendidos. Foi desenvolvido também uma API capaz de gerenciar as recomendações entre alunos e professores, sendo esse um dos frutos do projeto.

O presente trabalho está organizado da seguinte forma: A Seção 2 faz uma revisão dos principais conceitos utilizados no projeto; a Seção 3 abrange trabalhos relevantes para o presente estudo; os materiais e métodos empregados no desenvolvimento no estudo são descritos na Seção 4; os resultados obtidos são apresentados na Seção 5; a Seção 6 descreve o desenvolvimento de uma proposta de API integrada ao sistema de recomendação. As considerações finais são discutidas na Seção 7, seguidas das referências bibliográficas.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Inteligência Artificial

A Inteligência Artificial surgiu como uma solução para que as máquinas sejam capazes de tomar certas decisões no lugar de seus criadores, imitando sua capacidade de pensar, porém sem de fato fazê-lo. Por muito tempo, se achava que inteligência era a capacidade de raciocinar, hoje ela está mais ligada ao poder computacional. (TEIXEIRA, 2014). Sendo assim, a IA é uma área da computação que consiste em planejar, construir e validar sistemas inteligentes, ou que possuam características semelhantes à inteligência (SOWA, 2000). Os computadores recebem informações e a partir dessas entradas, se adaptam e geram soluções sem que sejam necessariamente programados para aquele problema específico. As pesquisas na área são bastante diversas, e atuam nos mais variados campos. Podemos citar entre eles a Visão Computacional, Processamento de Linguagem Natural, Raciocínio Automatizado e o Aprendizado de Máquina (GONDIM, 2022).

2.2 Sistemas de Recomendação

Os Sistemas de Recomendação surgiram para solucionar o problema da alta quantidade de dados, filtrando para o usuário apenas o conteúdo mais relevante dado à determinada situação (SILVA, 2020). Em outras palavras, consiste numa tecnologia analítica que calcula a probabilidade de um conteúdo ser selecionado por uma pessoa (PARK, 2011). Essas sugestões são baseadas no perfil do usuário, que por sua vez possui informações relevantes ao sistema de recomendação (AMORIM, 2010). Existem duas maneiras de extrair essas informações: a explícita e a implícita. Na explícita, o próprio usuário fornece essas informações, atribuindo notas ou avaliações a determinado conteúdo. Já na implícita são utilizadas funções que mapeiam ações do usuário, como histórico de uso e duração de acesso (BARBOSA, 2014).

Para gerar a recomendação, existem algumas técnicas de filtragem de informação, onde as abordagens mais famosas são a Filtragem por Conteúdo, Filtragem Colaborativa e Filtragem Híbrida (GONDIM, 2022). A Filtragem Colaborativa tenta realizar uma predição da relevância de um conteúdo baseando-se na avaliação de outros usuários (ADOMAVICIUS; TUZHILIN, 2005). Esta abordagem baseia-se no conceito que usuários que concordaram no passado irão concordar no futuro (IZBICKI; SANTOS, 2020), entretanto existe a possibilidade de não haver um padrão semelhante de interesse, ocasionado pela baixa amostragem de usuários ou avaliações (YANG; YANG; HU, 2016). Já a Filtragem por Conteúdo funciona de outra maneira, ela utiliza do histórico de preferências do usuário, buscando recomendar itens semelhantes a outros já bem avaliados por ele (SILVA, 2020). Nesta abordagem, existe o risco da superespecialização, onde o usuário fica limitado apenas aos itens semelhantes ao que já teve interesse anteriormente, trazendo a possibilidade de itens novos não serem recomendados (LÁZARO, 2010). Ambas as técnicas são essenciais para os sistemas de recomendação e possuem suas qualidades e situações onde deverão ser aplicadas. Através delas, é possível prever a relação entre diversos itens e

diversos usuários (BARBOSA, 2014). A Filtragem Híbrida consiste na junção de ambos os conceitos das filtragens anteriores, se aproveitando das vantagens e amenizando as desvantagens de cada uma (KIM et al., 2006).

2.3 Naive Bayes

O *Naive Bayes* consiste em um algoritmo de classificação que se baseia no teorema probabilístico de *Bayes*. O termo *naive* (inocente) vem da característica central do algoritmo, que considera todos os atributos como independentes entre si, dado o valor da classe. Essa suposição raramente é verdadeira nas aplicações reais, entretanto é muito conveniente, e pode levar a um excelente classificador (ZHANG, 2004).

Matematicamente, o algoritmo *Naive Bayes* estima a probabilidade de um dado desconhecido pertencer a uma classe específica com base nas probabilidades condicionais de cada atributo do objeto (ATHAYDE, 2023). O Teorema de *Bayes* em sua forma simplificada pode ser representado pela seguinte equação:

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

Onde:

$P(A)$ é a probabilidade a priori da classe (A) ocorrer;

$P(B)$ é a probabilidade a priori do preditor (B) ocorrer;

$P(A|B)$ é a probabilidade a posteriori da classe (A) condicional ao preditor (B);

$P(B|A)$ é a probabilidade a posteriori do preditor (B) condicional à classe (A).

Supondo que queremos classificar um e-mail como *spam* ou não *spam*, podemos utilizar como exemplo duas palavras “dinheiro” e “grátis” para calcular a probabilidade. Assim temos:

$P(S) = 0,6$ (60% dos e-mails são *spam*)

$P(NS) = 0,4$ (40% dos e-mails não são *spam*)

$P(S|“dinheiro”) = 0,8$ (80% dos e-mails *spam* contém a palavra “dinheiro”)

$P(NS|“dinheiro”) = 0,2$ (20% dos e-mails não *spam* contém a palavra “dinheiro”)

$P(S|“grátis”) = 0,9$ (90% dos e-mails *spam* contém a palavra “grátis”)

$P(NS|“grátis”) = 0,1$ (10% dos e-mails não *spam* contém a palavra “grátis”)

Dado um novo e-mail que possui ambas as palavras “dinheiro” e “grátis”, aplicamos o Teorema de Bayes e a suposição de independência condicional para calcular:

$$P(S|“dinheiro”, “grátis”) = P(S) * P(S|“dinheiro”) * P(S|“grátis”) = 0,6 * 0,8 * 0,9 = 0,432$$

$$P(\text{NSI}|\text{"dinheiro"}, \text{"grátis"}) = P(\text{NS}) * P(\text{NSI}|\text{"dinheiro"}) * P(\text{NSI}|\text{"grátis"}) = 0,4 * 0,2 * 0,1 = 0,008$$

$$P(\text{SI}|\text{"dinheiro"}, \text{"grátis"}) = 0,432 / (0,432 + 0,008) = 0,981$$

$$P(\text{NSI}|\text{"dinheiro"}, \text{"grátis"}) = 0,008 / (0,432 + 0,008) = 0,019$$

Portanto, o algoritmo classificou o novo e-mail como *spam*, com probabilidade de aproximadamente 98%. O exemplo foi adaptado de (GRUS, 2016).

3 TRABALHOS RELACIONADOS

Esta seção apresenta alguns trabalhos relacionados a Sistemas de Recomendação, onde cada trabalho possui uma abordagem diferente para solução de seus respectivos problemas.

O primeiro trabalho, com o título “Sistema de recomendação de plantio utilizando Aprendizado de Máquina” (ATHAYDE, 2023), realiza suas recomendações através de dados históricos de plantios, e suas respectivas condições ambientais, tais como temperatura, umidade e composição química do solo. O sistema trata esses dados para indicar a melhor cultura para ser plantada, dado as características do ambiente. Neste estudo foram aplicados cinco algoritmos, sendo eles *Decision Tree* (DT), *Random Forest* (RF), *K-Nearest Neighbors* (KNN), *Support Vector Machine* (SVM) e *Gaussian Naive-Bayes* (NB). O desempenho dos algoritmos foi mensurado através de quatro métricas: *Accuracy*, *Precision*, *Recall* e *F1 Score*. Dentre todos os modelos testados, RF e NB foram os que mais se destacaram nas métricas de desempenho. O trabalho resultou em uma aplicação web que coleta as informações necessárias do usuário e gera a recomendação.

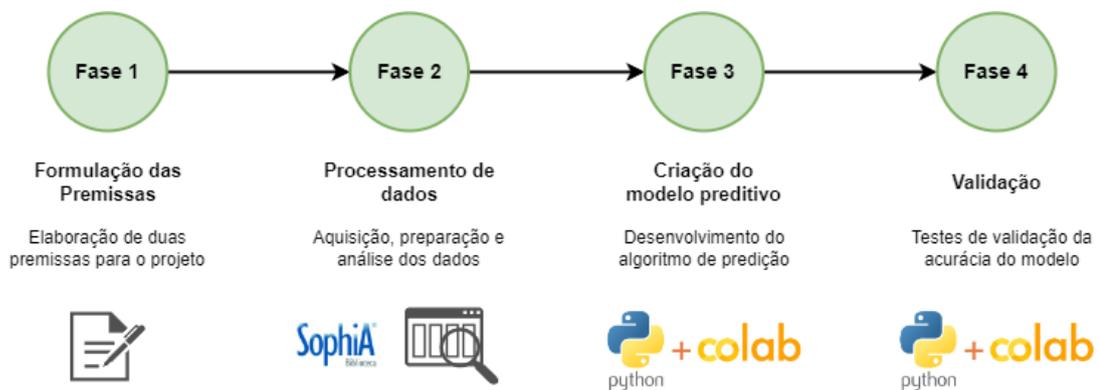
No trabalho de (GONDIM, 2019) foi realizada a proposta de um algoritmo genérico de filtro colaborativo para recomendação de itens entre usuários. A lógica da aplicação consiste em consultar no banco de dados usuários que avaliaram positivamente o mesmo item que o cliente, definindo assim os usuários semelhantes. Por seguinte, a aplicação busca itens que foram avaliados pelo usuário similar que ainda não foi avaliado pelo cliente da aplicação, e por fim, gera a recomendação deste item. A proposta geral do trabalho é ser genérico porém aplicável a diversos projetos de negócio.

Dentre os trabalhos sobre recomendação híbrida, é possível destacar o estudo de (SILVA, 2021). O trabalho combina técnicas de pré-filtragem contextual e filtragem híbrida para gerar recomendação de filmes, utilizando a base de dados *MovieLens*. A aplicação desenvolvida se comunica com o servidor, que por sua vez acessa o banco de dados de filmes. Para filtrar apenas os dados que se enquadram no contexto do usuário, foi realizada a técnica de pré-filtragem contextual e em seguida é gerado uma busca dos usuários semelhantes através do algoritmo KNN. A partir dos resultados obtidos, é aplicada a filtragem baseada em conteúdo, criando assim um algoritmo híbrido. O fluxo é finalizado com a recomendação sendo devolvida para o usuário através de uma API.

4 MATERIAIS E MÉTODOS

Esta seção irá detalhar as atividades desenvolvidas na elaboração do Sistema de Recomendação entre orientadores e alunos, com foco no curso de computação do IFCE- campus Aracati. Antes de dar início ao projeto, foi feita uma revisão bibliográfica sobre Sistemas Recomendativos e suas diversas aplicações. Esta pesquisa foi realizada em diversos artigos no estado da arte, onde foram investigados quais algoritmos e técnicas poderiam ser aplicadas ao problema explorado. O estudo proporcionou um alicerce de conhecimento das abordagens utilizadas nos sistemas de recomendação. A Figura 1 apresenta fluxo de elaboração do projeto, bem como as respectivas tecnologias utilizadas.

Figura 1 – Fases da metodologia adotada para o projeto



Fonte: Autoria própria

4.1 Formulação das Premissas

Na primeira fase do projeto, tratou-se de formular duas premissas fundamentais que desempenharam um papel importante no desenvolvimento geral da pesquisa.

4.1.1 Primeira premissa

A primeira premissa parte do princípio de que professores que já foram orientadores de diversos alunos num determinado tema, são os ideais para orientar aquele assunto. Essa abordagem fundamenta-se na crença de que ao examinarmos os registros históricos e repetirmos padrões passados, temos a possibilidade de alcançar o mesmo êxito. Entretanto, essa teoria nem sempre é verdadeira no mundo real, sendo um efeito colateral imediato dessa premissa a exclusão de docentes que não estão presentes na base histórica, pois o algoritmo não é capaz de “enxergar” além do passado. As etapas buscaram analisar e comprovar se é possível aplicar essa premissa ao universo de trabalhos defendidos estudado. Os resultados da premissa são detalhados na Seção 5.2.

4.1.2 Segunda premissa

Já a segunda premissa traz uma visão diferente do problema. Ela fundamenta-se no princípio de que docentes com mais tempo de ensino em uma determinada matéria tendem a ter mais conhecimento sobre o assunto, o tornando um excelente candidato a orientador. A pesquisa inicial teve como objetivo investigar a influência desse tempo de experiência do professor na sua seleção como orientador em projetos relacionados à disciplina. Para comprovar essa premissa, as fases subsequentes do projeto também foram impactadas, em especial o algoritmo utilizado. Além disso, uma permutação de pesos foi feita para avaliar o desempenho de ambas as premissas. Os resultados obtidos também estão descritos na Seção 5.2.

4.2 Processamento de Dados

Para que o sistema consiga realizar previsões, torna-se importante possuir dados históricos para calcular a probabilidade do orientador ser o ideal para o projeto do aluno. Esta fase do projeto buscou coletar, preparar e analisar os dados, seguindo as premissas definidas na etapa anterior.

4.2.1 Aquisição dos Dados

Primeiramente foi realizada a coleta dos dados necessários para montar a base histórica de TCCs defendidos, além de separar quais classes (orientadores) que serão recomendadas. O IFCE mantém um registro histórico de todos Trabalhos de Conclusão de Curso já defendidos em sua biblioteca virtual SophiA, assim os 83 projetos (atualizado por último em 2023) dos alunos concludentes do curso de computação do campus Aracati foram coletados para servir como base de dados de teste. Os artigos possuem informações essenciais para a recomendação, tais como nome do orientador, coorientador e temática do estudo.

4.2.2 Preparação dos Dados

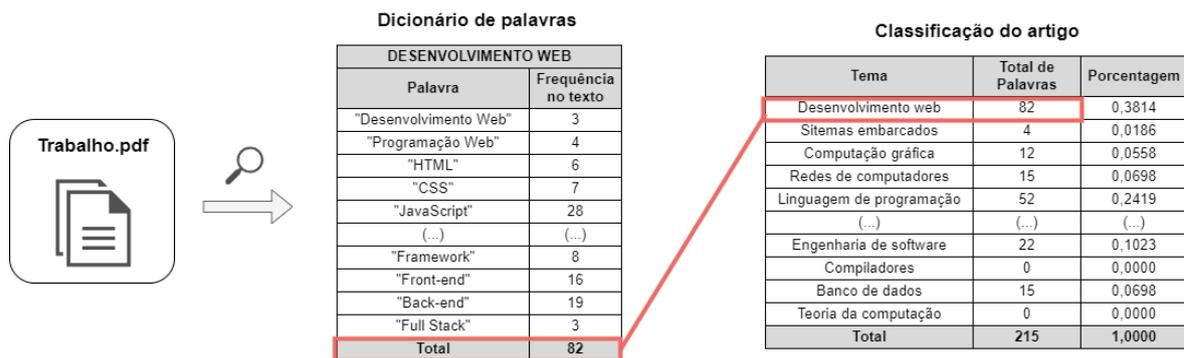
É na preparação dos dados onde é definido os atributos classificadores do sistema (temas dos trabalhos). Esta parte é importante, pois estas serão as características que todos os TCCs e professores possuirão. Para diversificar a base de dados, os orientadores e coorientadores possuem a mesma relevância em termos de participação em um trabalho, além disso, os orientadores que não integram o corpo docente do IFCE foram excluídos, contribuindo para a redução de ruídos no conjunto de dados. O curso de computação do campus Aracati teve um total de 62 professores desde sua criação, onde 16 desses docentes já foram orientadores.

A classificação dos temas foi feita utilizando um *script*¹ que realiza uma busca automatizada no texto de cada trabalho, através de um dicionário em Python. Nesses trabalhos foram medidas as frequências relativas (isto é, a proporção de vezes que a palavra aparece no texto) de

¹ Disponível em: <https://github.com/Maironsantana/TCC-Sistema-de-Recomendacao-Professores>

50 palavras para cada tema. A biblioteca PyPDF2² foi uma ferramenta essencial para o processo, pois através dela foi possível manipular e contar as frequências relativas dentro do arquivo PDF dos trabalhos. Desta forma é possível obter um valor de relevância associado a um determinado tema. A Figura 2 ilustra o funcionamento do algoritmo desenvolvido para classificar o TCC. Ao final do processo, os três temas com maiores valores de porcentagem serão definidos como os temas do artigo.

Figura 2 – Processo de classificação dos TCCs



Fonte: Autoria própria

Para o presente trabalho, dois *datasets* de trabalhos defendidos foram criados, cada um com uma divisão de temas diferentes. Ambas foram analisadas e posteriormente testadas com o algoritmo desenvolvido. A primeira classificação de temas foi feita com base na matriz curricular do curso, desta forma o tempo de docência de cada professor nas matérias poderiam ser facilmente levadas em consideração no cálculo de probabilidade. Inicialmente todas matérias do curso entraram na separação, porém percebeu-se uma baixa (ou nenhuma) adesão de artigos à matérias específicas e introdutórias (como Português Instrumental por exemplo). Por isso foram separadas 24 disciplinas mais expressivas nos artigos para servirem como temas durante a classificação dos artigos. Podemos visualizar as matérias selecionadas na Figura 3.

Figura 3 – Matérias mais expressivas em TCCs defendidos

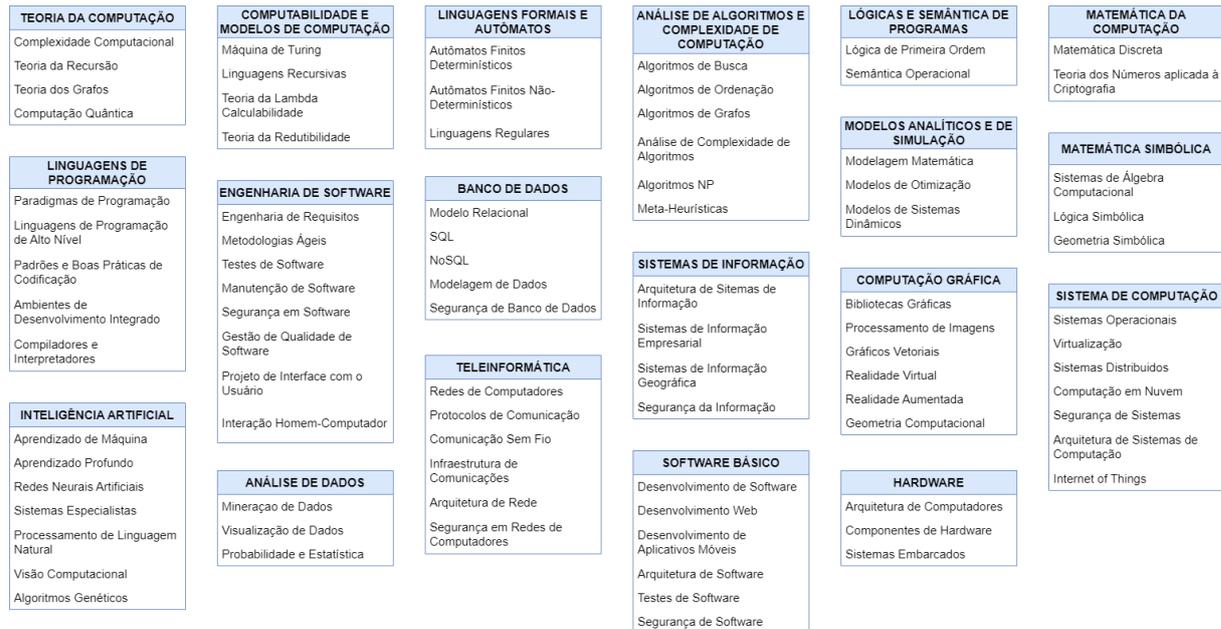
PRINCIPAIS DISCIPLINAS	
ANÁLISE E PROJETO DE SISTEMAS	PROBABILIDADE E ESTATÍSTICA
BANCO DE DADOS	PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS
COMPILADORES	REDES DE COMPUTADORES
COMPUTAÇÃO GRÁFICA	SEGURANÇA DA INFORMAÇÃO
CONSTRUÇÃO E ANÁLISE DE ALGORITMOS	SISTEMAS DE INFORMAÇÕES GEOGRÁFICAS
DESENVOLVIMENTO WEB	SISTEMAS EMBARCADOS
DESIGN WEB	TEORIA DA COMPUTAÇÃO
ENGENHARIA DE SOFTWARE	MATEMÁTICA DISCRETA
INFORMÁTICA NA EDUCAÇÃO	PROJETO DE INFRAESTRUTURA DE REDES
INTELIGÊNCIA ARTIFICIAL	LINGUAGEM DE PROGRAMAÇÃO
INTERAÇÃO HOMEM-COMPUTADOR	LINGUAGENS FORMAIS E AUTÔMATOS
LINGUAGEM DE PROGRAMAÇÃO	PROGRAMAÇÃO ORIENTADA A OBJETOS

Fonte: Autoria própria

² Disponível em: <https://pypi.org/project/PyPDF2/>

Na segundo *dataset*, os temas foram separados pelas subáreas de conhecimento da CAPES relacionadas a computação (CAPES, 2023). Para cada uma das 19 subáreas, foi feita uma pesquisa em periódicos e revistas científicas em busca dos temas mais abordados na subárea da computação. Foram definidos entre 3 e 6 tópicos para cada subárea, resultando em 86 temas. A Figura 4 mostra uma listagem de todos os tópicos utilizados no segundo estudo.

Figura 4 – Tópicos organizados em subáreas da computação

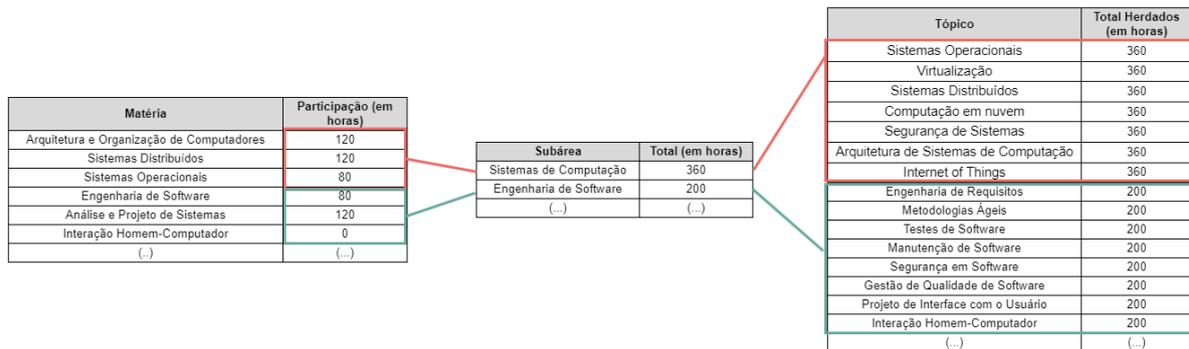


Fonte: Autoria própria

Adotando essa abordagem, expandimos o espectro de temas que o sistema de recomendação pode trabalhar. Contudo, não é mais tão simples medir o impacto do tempo de docência do professor nos temas da computação. Para superar esse desafio, foi realizado um mapeamento dos temas em relação às disciplinas do curso. O estudo conduzido por (FELIPE; GALLINDO; GOMES, 2023), que colaborou com os coordenadores dos cursos do IFCE para mapear as disciplinas nas subáreas da computação, desempenhou um papel fundamental nesse processo. O tempo de ensino nas disciplinas de uma determinada subárea foram somadas e herdadas pelos respectivos tópicos. A Figura 5 ilustra o processo de mapeamento.

Após a definição dos temas, procedemos para a elaboração dos dicionários e a utilização do *script* classificador desenvolvido. Como resultado, foram gerados dois (*datasets*), cada um contendo a pontuação da frequência relativa para cada trabalho dentro dos temas preestabelecidos. Essa abordagem permitiu o cálculo da porcentagem de contribuição dos orientadores para cada tema, sendo essa uma etapa essencial na criação de uma tabela de probabilidade empregada pelo sistema de recomendação.

Figura 5 – Processo de mapeamento do tempo de cada professor nos tópicos



Fonte: Autoria própria

4.2.3 Análise dos dados

A partir da primeira modelagem dos dados, já se torna possível extrair informações relevantes acerca do ecossistema educacional do IFCE - campus Aracati. Um exemplo inicial é a apresentação de um panorama abrangente dos temas que concentram a maior quantidade de trabalhos na área, o que reflete o interesse predominante dos estudantes de Computação no campus Aracati. Por outro lado, essa análise também identifica os temas com uma representação mais reduzida de trabalhos, fornecendo *insights* valiosos para a administração do curso no sentido de atrair alunos para esses tópicos específicos. Este esforço contribui significativamente para a diversidade de projetos em todas as áreas da computação. Adicionalmente, é possível obter a frequência de orientação de cada professor, sendo esse um dado adicional de possível interesse para a coordenação do curso. Todos os resultados e gráficos gerados pela análise podem ser visualizados na Seção 5.1.

4.3 Criação do modelo preditivo

Para o presente estudo, optou-se pela utilização do modelo preditivo *Naive Bayes*. Este algoritmo demonstrou eficácia significativa na predição de problemas multiclases, conforme observado na revisão bibliográfica (LUNARDI; VITERBO; BERNARDINI, 2016). Um dos primeiros desafios ao trabalhar com este algoritmo surgiu devido a uma quantidade expressiva de valores nulos no dataset. O modelo de *Naive Bayes*, por ser sensível a atributos nulos, acaba subestimando ou superestimando a probabilidade de um novo dado pertencer a uma determinada classe. Essa ocorrência é facilmente observável na base de dados utilizada, tendo em vista que raramente um professor orientou todos os temas disponíveis. Para lidar com esse problema, um pré-processamento foi feito no dataset, adicionando valores fictícios à base de dados, para evitar multiplicação por zero. Foi tomado o cuidado de adicionar valores pouco expressivos, evitando interferir no cálculo de probabilidade. O código foi todo desenvolvido utilizando a ferramenta *Google Colab*³.

³ Disponível em: <https://github.com/Maironsantana/TCC-Sistema-de-Recomendacao-Professores>

Para que o algoritmo consiga trabalhar com ambas premissas, ele sofreu uma adaptação para contemplar tanto a porcentagem de participação como orientador quanto tempo de ensino na matéria definida (conforme discutido na Seção 4.1). O algoritmo irá calcular a probabilidade de cada classe (professor) através da multiplicação dos valores de probabilidade a posteriori dos preditores (temas), juntamente com sua probabilidade a priori como orientador. Para verificarmos o impacto do tempo de docência, é feito também uma multiplicação pela participação relativa em horas como professor no tema. A adaptação na fórmula matemática do algoritmo pode ser observado na Figura 6:

Figura 6 – Adaptação no algoritmo *Naive Bayes*

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \longrightarrow P(A|B) = \frac{P(Bo|A) * P(Bh|A) * P(A)}{P(B)}$$

Fonte: Autoria própria

Onde:

$P(A)$ é a probabilidade a priori do professor (A);

$P(B)$ é a probabilidade a priori do tema (B);

$P(A|B)$ é a probabilidade a posteriori do professor (A) condicional ao tema (B);

$P(Bo|A)$ é a participação como orientador no tema (B) do professor (A).

$P(Bh|A)$ é a participação em horas ensinando o tema (B) do professor (A).

4.4 Validação

O algoritmo foi treinado e testado com os mesmos dados, utilizando a estratégia *K-fold Cross Validation*. A técnica divide o conjunto de dados em K partes iguais, chamadas *folds* (subconjuntos). O modelo é treinado K vezes, cada vez utilizando K-1 *folds* como conjunto de treinamento e o *fold* restante como conjunto de validação. As divisões são feitas mantendo a proporção de aparição dos professores dentro dos subconjuntos. Para o estudo foram definidos 4 *folds*, conforme apresentado na Figura 7:

Figura 7 – 4-fold Cross Validation

	FOLD 1	FOLD 2	FOLD 3	FOLD 4
ITERAÇÃO 1	TREINO			TESTE
ITERAÇÃO 2	TREINO		TESTE	TREINO
ITERAÇÃO 3	TREINO	TESTE	TREINO	
ITERAÇÃO 4	TESTE	TREINO		

Fonte: Adaptado de (WASNIK, 2020)

Ao concluir cada iteração, é obtido um valor de acurácia, e ao final é tirado uma média geral das 4 iterações, obtendo uma acurácia total do round de teste.

Para observar o desempenho de ambas as premissas, foram feitos 5 testes variando pesos de horas (PH) e os pesos de orientação (PO) onde, para cada rodada de testes, houve uma configuração diferente de pesos aplicados aos valores de probabilidade ($P(B|A)$ e $P(B|A)$ respectivamente). A permutação de pesos ocorreu da seguinte forma:

- Primeiro Round: Peso Horas: 1,0 | Peso Orientação: 0,0
- Segundo Round: Peso Horas: 0,25 | Peso Orientação: 0,25
- Terceiro Round: Peso Horas: 0,5 | Peso Orientação: 0,5
- Quarto Round: Peso Horas: 0,25 | Peso Orientação: 0,75
- Quinto Round: Peso Horas: 0,0 | Peso Orientação: 1,0

Todos os resultados obtidos na validação podem ser observados na Seção 5.2.

5 RESULTADOS

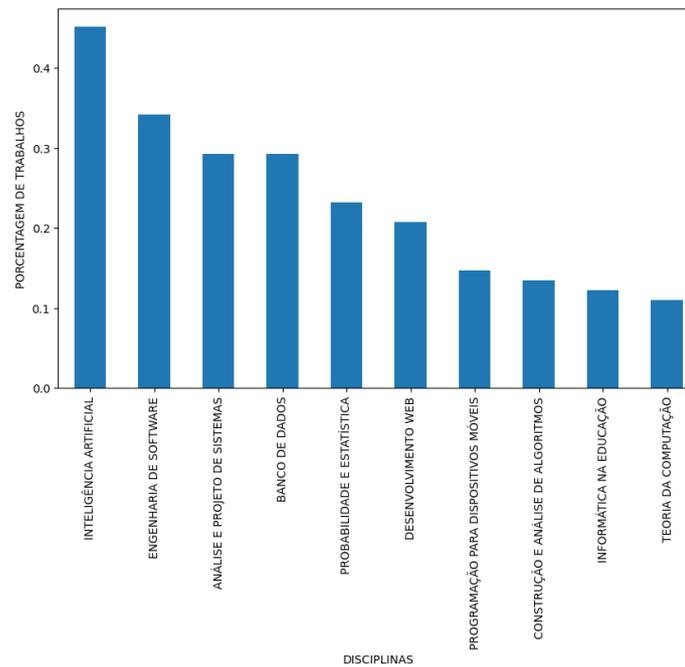
Nesta seção são apresentados os resultados obtidos durante o estudo sobre sistemas de recomendação no contexto de alunos e orientadores do IFCE - campus Aracati.

5.1 Análise de dados

Alguns dados interessantes foram obtidos através do *script* para contagem de frequência relativa de palavras. Trabalhar com a base de dados de TCCs proporcionou a extração de informações relacionadas ao panorama histórico de trabalhos no campus. Na Figura 8 é possível observar as 10 disciplinas de Bacharelado em Ciência da Computação com mais trabalhos relacionados à matéria. Os valores dizem respeito a porcentagem de trabalhos classificados naquele tema.

Vale ressaltar a característica intrínseca dos TCCs de possuírem múltiplas temáticas, ou seja, cada trabalho pode possuir mais de uma matéria relacionada. Podemos observar como a temática de Inteligência Artificial é muito dominante e está presente em cerca de 45,12% dos trabalhos. Outras matérias com grande expressividade de trabalhos são Engenharia de Software (34,15%), Análise e Projeto de Sistemas (29,26%), Banco de Dados (29,26%) e Probabilidade e Estatística (23,17%). Por outro lado, matérias relacionadas à teoria da computação tendem a ter uma menor adesão, como por exemplo Compiladores e Linguagens Formais e Autômatos. Uma consequência direta da utilização das matérias do curso como atributo preditor foi a generalização de tópicos específicos, assim perdendo muita informação relacionada ao TCC. Por exemplo, um trabalho sobre redes neurais seria classificado como semelhante a um trabalho de processamento de linguagem natural, pois ambos estariam dentro de “Inteligência Artificial”. Embora sejam áreas que podem coexistir num mesmo trabalho, são intrinsecamente diferentes e uma não está necessariamente ligada à outra.

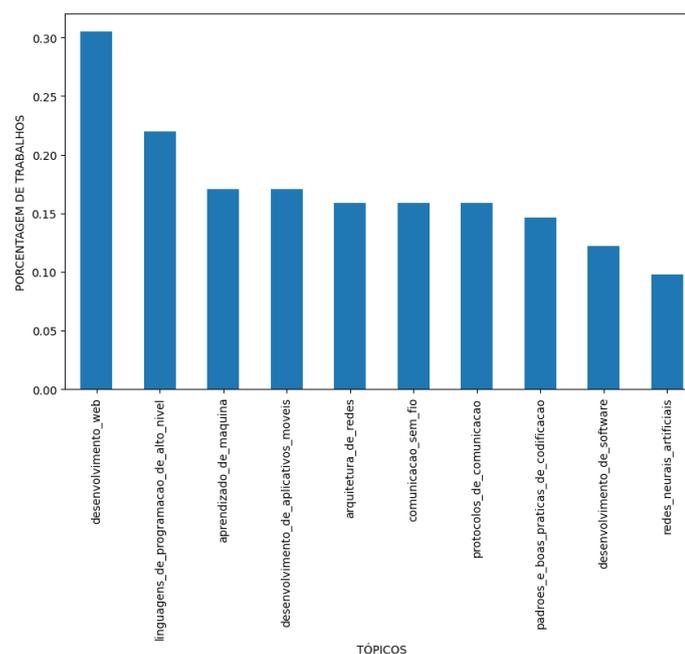
Figura 8 – Gráfico das 10 matérias com mais trabalhos relacionados



Fonte: Autoria Própria

Ao olharmos para esses dados sob uma perspectiva diferente (agora classificados nos tópicos de computação), surge um novo panorama. Uma primeira vantagem da não vinculação estrita às disciplinas do curso é a flexibilidade do sistema para ser aplicado em outros cursos dentro da área de computação. Podemos observar na Figura 9, os dez tópicos com maior pontuação na contagem de frequências relativas de palavras.

Figura 9 – Gráfico dos 10 tópicos mais abordados em trabalhos

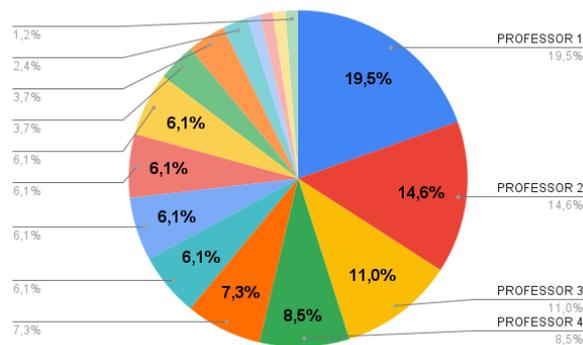


Fonte: Autoria Própria

É interessante notar que devido a decomposição de sub-áreas em tópicos, a expressividade de algumas matérias dominantes do gráfico anterior foram diluídas em temas menores, como por exemplo Inteligência Artificial e Engenharia de Software. Por outro lado, alguns temas se destacaram da estatística anterior, como Desenvolvimento Web, Desenvolvimento de Aplicativos Móveis e Linguagens de Programação. É relevante ressaltar a escassez de estudos abordando os temas de Segurança de Software, Segurança de Sistemas e Segurança de Infraestrutura de Redes, apesar do elevado interesse nessas áreas temáticas. Isso pode significar um sintoma maior, de que os alunos não estão interessados na segurança de seus projetos.

O gráfico da Figura 10 apresenta a distribuição das porcentagens de participação dos professores como orientadores de TCCs, sem revelar identidades específicas, em relação a todos os trabalhos já apresentados. A estatística traz todos os professores que já orientaram, inclusive os que não estão mais ativos no campus, onde cada porção representa um docente, contendo sua respectiva porcentagem. A visualização destaca uma significativa disparidade nas contribuições, evidenciando que alguns professores possuem uma participação substancialmente mais elevada em comparação com outros. Um único docente possui cerca de 19,5% de participação como orientador, enquanto os demais mantêm valores próximos ou inferiores a 10%. Essa grande disparidade de participação impacta diretamente no modelo preditivo, dificultando previsões mais precisas. A principal suposição do motivo da disparidade entre esses valores é de que alunos tendem a concentrar suas escolhas de orientadores em professores que ministram bolsas e gerenciam laboratórios de desenvolvimento.

Figura 10 – Porcentagem de participação dos docentes como orientadores



Fonte: Autoria Própria

5.2 Testes do Algoritmo

A seguir, temos os resultados obtidos com o algoritmo *Naive Bayes*, aplicando a técnica *4-fold Cross Validation*. A Figura 11, possui uma tabela representando os resultados obtidos nos testes com o *dataset* de disciplinas variando os pesos (Abordado na Seção 4.4). O valor de acurácia de cada incremento do *Cross Validation* foi usado para calcular a média de acurácia (MA) do modelo. O estudo trouxe resultados para até 5 orientadores recomendados. É interessante notar que ao considerar apenas o tempo de docência na disciplina para gerar a recomendação,

o algoritmo obteve pior desempenho. Este resultado está diretamente relacionado ao dado da Figura 10 que demonstra um grande desequilíbrio da base de dados em relação à porcentagem de participação. Isso indica que, nos Trabalhos de Conclusão de Curso de computação do campus Aracati, é bastante comum docentes orientarem trabalhos fora do seu leque de disciplinas ministradas. Dito isto, a distribuição de pesos que trouxe o melhor valor de acurácia foi com peso em horas nulo e o peso de orientação 1,0.

Figura 11 – Tabela de acurácia do algoritmo com o primeiro *dataset* (Disciplinas)

Qtd. Orientadores Recomendados	PH: 1,0 PO: 0,0	PH: 0,75 PO: 0,25	PH: 0,5 PO: 0,5	PH: 0,75 PO: 0,25	PH: 0,0 PO: 1,0
	MA	MA	MA	MA	MA
1	0,2214	0,2822	0,2823	0,2822	0,3059
2	0,3190	0,3923	0,3923	0,3923	0,4518
3	0,4161	0,4643	0,4643	0,4643	0,5625
4	0,5005	0,5493	0,5493	0,5493	0,6226
5	0,5737	0,6351	0,6351	0,6351	0,7321

Fonte: Autoria Própria

A tabela presente Figura 12 apresenta os resultados dos mesmos testes aplicados com o *dataset* de tópicos da computação.

Figura 12 – Tabela de acurácia do algoritmo com o segundo *dataset* (Tópicos)

Qtd. Orientadores Recomendados	PH: 1,0 PO: 0,0	PH: 0,75 PO: 0,25	PH: 0,5 PO: 0,5	PH: 0,75 PO: 0,25	PH: 0,0 PO: 1,0
	MA	MA	MA	MA	MA
1	0,2089	0,1952	0,1952	0,1952	0,3060
2	0,3548	0,4024	0,4024	0,4024	0,5000
3	0,4268	0,5369	0,5369	0,5369	0,6220
4	0,5250	0,6589	0,6589	0,6589	0,7185
5	0,6470	0,6952	0,6952	0,6952	0,7554

Fonte: Autoria Própria

Podemos observar que o algoritmo nesse contexto teve um desempenho bem semelhante ao anterior. Mesmo não sendo ainda um resultado ótimo, já é aceitável para o problema proposto. Por isso, é necessário ressaltar que ao trabalhar com recomendação em um contexto como esse é importante não depender apenas da base histórica, pois o algoritmo não terá capacidade de “enxergar” além do passado, correndo o risco de sempre recomendar os mesmos orientadores. Aqui entra a necessidade de se trabalhar também com outros tipos de recomendação, para ampliar o leque de possibilidades para o usuário do sistema. Na seção a seguir é apresentado uma proposta de uma API que busca solucionar o problema descrito.

6 DESENVOLVIMENTO DA API

A solução adotada consiste numa API capaz de gerenciar todo o fluxo de recomendação. Para enfrentar o problema da “exclusão” de docentes com poucos ou nenhum trabalho orientado, a aplicação foi pensada para realizar recomendações aos usuários seguindo três parâmetros: disponibilidade, afinidade e relevância.

6.1 Recomendação por Disponibilidade

A recomendação mais simples, consiste num valor atribuído a cada orientador, representando o número de vagas disponíveis para orientação. O docente ficará sempre ativo enquanto possuir vagas de orientação, porém quando este valor for zero, ele será listado como indisponível, não sendo recomendado para o aluno.

6.2 Recomendação por Afinidade

Essa forma de recomendação será feita utilizando dados que o próprio orientador fornece dos temas que ele tem mais interesse por orientar. Um administrador tem a responsabilidade de cadastrar esses temas de preferência no banco de dados. Assim, quando um aluno busca um tema, as recomendações de afinidade são feitas listando todos os professores que marcaram preferência pelo tema, ordenando pelo que possuir mais vagas de orientação.

6.3 Recomendação por Relevância

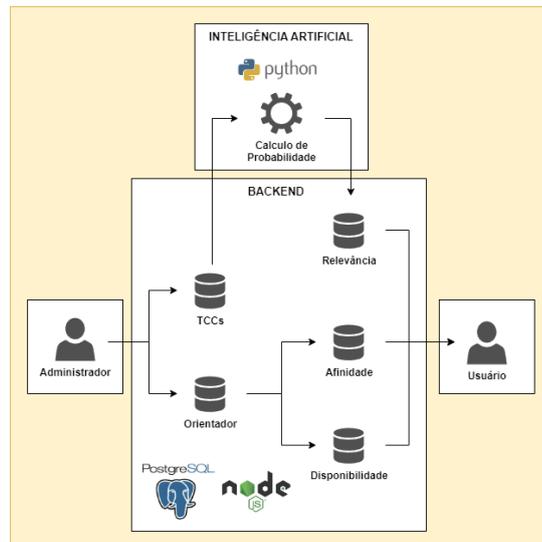
A recomendação por relevância será calculada através do módulo de inteligência artificial, utilizando a segunda premissa estudada neste trabalho. A premissa foi selecionada principalmente pela ampla variedade de temas, flexibilizando o sistema para os mais diversos cursos de computação. O módulo de inteligência artificial fará todo o cálculo de predição de cada professor para o tema selecionado. A IA retorna a pontuação de relevância (isto é, o valor de probabilidade) de cada orientador para ser cadastrada no banco de dados. Vale lembrar que o sistema somente recomenda um orientador se este possuir vagas disponíveis para orientação.

6.4 Fluxo da aplicação e tecnologias utilizadas

A aplicação foi projetada para manter o sistema sempre relevante, através da inserção de novos dados por um usuário administrador. Este super usuário tem a responsabilidade de cadastrar e editar os orientadores, além de inserir novos TCCs da base de dados que serão utilizados para melhorar a recomendação da IA. A Figura 13 apresenta uma visão geral do fluxo da aplicação.

Foi escolhido como linguagem principal do sistema o Javascript (JS). Sua facilidade de aprendizado e uso, juntamente com a capacidade de reutilização de código e suporte a bibliotecas e frameworks populares, tornam-na uma boa ferramenta. Além disso, foi incluído o TypeScript (TS) no projeto, visando a detecção antecipada de erros durante a fase de desenvolvimento. Isso

Figura 13 – Fluxo da aplicação

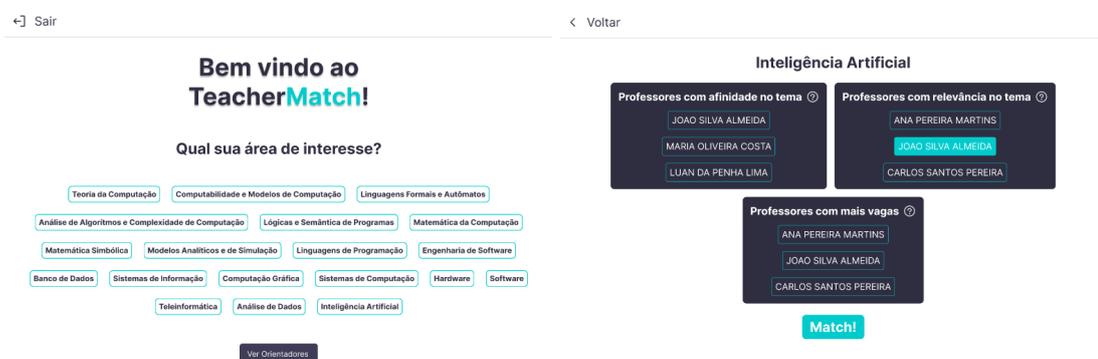


Fonte: Autoria Própria

contribui para reforçar a robustez e confiabilidade do código, além de aprimorar a segurança na transmissão de dados dentro da aplicação. Toda estrutura de backend, que é responsável pelas regras de negócio da aplicação, foi desenvolvido com o framework Node.js, juntamente com o Prisma, um framework especializado em mapeamento objeto-relacional (ORM). Para o banco de dados relacional optamos por implementar o PostgreSQL, utilizando um container Docker para sua implantação. Essa escolha oferece uma série de benefícios permitindo aproveitar a flexibilidade e escalabilidade proporcionadas pelos containers Docker.

Além disso, foi feito o protótipo de telas da aplicação. A Figura 14a representa a tela de seleção do tema de interesse do aluno, e a Figura 14b representa a recomendação propriamente dita, onde o usuário recebe uma listagem dos professores com melhor pontuação para cada tipo de recomendação no tema escolhido. Após selecionado o professor, o aluno realizará o pedido de *match*, que poderá ser visualizado e aceito por um usuário administrador.

Figura 14 – Protótipo de telas



(a) Tela de seleção do tema

(b) Tela de recomendação

Fonte: Autoria Própria

7 CONCLUSÃO

O presente estudo apresentou um sistema recomendativo entre alunos e professores de computação com intuito de ajudar alunos a encontrarem orientadores para seus projetos. Para atingir este objetivo, foi feito um trabalho de análise de dados sobre o ecossistema de TCCs defendidos do IFCE - campus Aracati. Essa investigação foi capaz de identificar os temas que concentram a maior quantidade de trabalhos na área, o que reflete o interesse predominante dos estudantes de Computação no campus Aracati. Por outro lado, também trouxe à tona os temas com uma adesão mais reduzida, fornecendo insights valiosos para a administração do curso no sentido de atrair alunos para esses tópicos específicos. Este esforço contribui significativamente para o aumento da diversidade de projetos em todas as áreas da computação no campus Aracati. Além disso, estudos e adaptações foram feitos com o algoritmo *Naive Bayes*, a fim de montar um modelo preditivo capaz de recomendar os orientadores ideais para os alunos. Os resultados de predição ressaltam a importância de não depender apenas da base histórica para gerar a recomendação.

Como um resultado extra, o trabalho também trouxe um projeto de API⁴ capaz de gerenciar todo o sistema de recomendação, além de trazer um leque maior de possibilidades para o usuário, trazendo para o sistema a preferência pessoal do professor e sua disponibilidade para orientação. A API foi projetada não somente para gerenciar os *matches*, como também para incrementar a base de dados por meio de novas inserções de artigos. Com este modelo de aplicação o sistema ficará sempre vivo, necessitando apenas de atualizações nos dados a cada novo ciclo de orientação.

Dentre os trabalhos futuros, espera-se desenvolver todo o *frontend* da aplicação web, bem como a integração com a API construída no presente estudo. Nesse processo, poderá ser implementado diversas medidas para reforçar a segurança do software, acompanhadas de testes de ponta a ponta para avaliar o desempenho antes da fase de produção. Além disso, também pretende-se realizar testes práticos com as futuras turmas de Trabalho de Conclusão de Curso (TCC) do Bacharelado em Ciência da Computação do IFCE - campus Aracati.

O projeto também abre novas portas para estudos e melhorias no sistema recomendativo. No intuito de otimizar a classificação da base histórica de TCCs, poderá ser realizado um trabalho de ampliação do dicionário de palavras dos tópicos, além da adoção de uma abordagem que atribua pesos diferenciados a cada termo. Adicionalmente, é possível aprimorar a separação dos tópicos, identificando novos temas emergentes ou tópicos redundantes. Destaca-se também a exploração de soluções para lidar com a natureza interdisciplinar de áreas específicas, como é o caso da Internet das Coisas (IoT) e entre outros exemplos. Além disso, pode ser explorado outros algoritmos de predição para o problema, como por exemplo o K-Nearest-Neighbors (KNN), podendo trazer uma visão diferente ao problema enfrentado.

REFERÊNCIAS

⁴ Disponível em: <https://github.com/Maironsantana/tcc-api-sistema-de-recomendacao>

ADOMAVICIUS, G.; TUZHILIN, A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. **IEEE Transactions on Knowledge and Data Engineering**, v. 17, n. 6, p. 734–749, 2005.

AMORIM, J. S. de. **UM SERVIÇO DE RECOMENDAÇÃO INTER-APLICAÇÕES BASEADO EM FILTRAGEM COLABORATIVA**. 2010. Escola Politécnica de Pernambuco – Universidade de Pernambuco.

ATHAYDE, M. P. Sistema de recomendação de plantio utilizando aprendizado de máquina. Universidade Federal de Uberlândia, 2023.

BARBOSA, C. E. M. **Estudo de Técnicas de Filtragem Híbrida em Sistemas de Recomendação de Produtos**. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 3 2014.

CANZIANI, I. F. S. et al. **Evasão dos cursos de pós-graduação lato sensu (2010-2014) da Universidade do Sul de Santa Catarina/UNISUL-Campus Sul, Tubarão/SC**. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis, 2015.

CAPES. **Tabela de Áreas de Conhecimento/Avaliação**. 2023. Disponível em: <<https://www.gov.br/capes/pt-br/aceso-a-informacao/acoes-e-programas/avaliacao/instrumentos/documentos-de-apoio-1/tabela-de-areas-de-conhecimento-avaliacao>>.

COOKE, P.; URANGA, M. G.; ETXEBARRIA, G. Regional innovation systems: Institutional and organisational dimensions. **Research policy**, Elsevier, v. 26, n. 4-5, p. 475–491, 1997.

DANA, S. **A importância da educação para o crescimento econômico**. Disponível em: <https://g1.globo.com/economia/blog/samy-dana/post/importancia-da-educacao-para-o-crescimento-economico.html>, 2017. Acesso em: 24 de março de 2023.

FAUSTINO-FERBER, C. M. H. A. P. Motivos que levam os alunos à evasão em cursos de pós-graduação lato sensu em instituição pública de educação. **Revista Labor, Fortaleza**, v. 2, n. 26, p. 31–55, 12 2021.

FELIPE, M. d. S.; GALLINDO, É. d. L.; GOMES, L. B. S. Dimensionamento da força de trabalho docente: uma abordagem computacional baseada nas áreas de conhecimento dos cursos oferecidos. In: SBC. **Anais do VIII Congresso sobre Tecnologias na Educação**. [S.l.], 2023. p. 282–291.

GONDIM, R. C. **Proposta de um sistema de recomendação baseado em preferências do usuário**. Dissertação (B.S. thesis) — Universidade Federal do Rio Grande do Norte, 2019.

GONDIM, R. dos S. **Mecanismo de recomendação personalizada de rotas utilizando algoritmos genéticos no contexto do turismo 4.0**. Dissertação (Mestrado) — UFERSA, 8 2022.

GRUS, J. **Data Science do Zero**. Alta Books, 2016. ISBN 9788550803876. Disponível em: <<https://books.google.com.br/books?id=2LZwDwAAQBAJ>>.

GUIMARÃES, J. de C.; SOBRINHO, F. D. da S. Fatores facilitadores e dificultadores à construção do tcc. **Revista Brasileira de Administração Científica**, v. 11, n. 3, p. 82–99, 2020.

IFCE. **Em Números - IFCE**. Disponível em: <https://ifceemnumeros.ifce.edu.br/>, 2023. Acesso em: 2 de outubro de 2023.

IZBICKI, R.; SANTOS, T. dos. **Aprendizado de máquina: uma abordagem estatística**. Rafael Izbicki, 2020. ISBN 9786500024104. Disponível em: <<https://books.google.com.br/books?id=6O8OEAAAQBAJ>>.

JUNIOR, C. A. da C. et al. Representações sociais de emoções e sentimentos na elaboração de trabalho de conclusão de curso. **Revista de Contabilidade e Controladoria**, v. 14, n. 1, 2022.

KIM, B. M. et al. A new approach for combining content-based and collaborative filters. **Journal of Intelligent Information Systems**, Springer, v. 27, p. 79–91, 2006.

LÁZARO, A. d. S. Análise e seleção de algoritmo de filtragem de informação para solução do problema cold-start item. **Monografia de Graduação, Departamento de Ciência da Computação, Universidade Federal de Lavras**, 2010.

LUNARDI, A.; VITERBO, J.; BERNARDINI, F. Análise de sentimentos utilizando técnicas de classificação multiclasse. **Sociedade Brasileira de Computação**, 2016.

PARK, D. H. A literature review and classification of recommender systems on academic journals. **Journal of intelligence and information systems**, v. 17, n. 1, p. 139–152, 2011.

SILVA, L. M. d. Sistema de recomendação híbrido utilizando as técnicas de filtragem colaborativa e baseada em conteúdo. 2021.

SILVA, L. M. da. **Sistema de recomendação híbrido utilizando as técnicas de filtragem colaborativa e baseada em conteúdo**. 2020. UNESC.

SOWA, J. **Knowledge Representation: Logical, Philosophical, and Computational Foundations**. Brooks/Cole, 2000. (Computer Science Series). ISBN 9780534949655. Disponível em: <<https://books.google.com.br/books?id=dohQAAAAMAAJ>>.

TAKAHASHI, M. M.; JR, R. H. Estudo comparativo de algoritmos de recomendação. **Universidade de São Paulo**, 2015.

TEIXEIRA, J. de F. **Inteligência artificial**. Paulus Editora, 2014. (Como ler filosofia). ISBN 9788534936842. Disponível em: <<https://books.google.com.br/books?id=79q5DAAAQBAJ>>.

VICARI, R. M. Influências das tecnologias da inteligência artificial no ensino. **Estudos Avançados**, SciELO Brasil, v. 35, p. 73–84, 2021.

WASNIK, A. **K-Fold Cross-Validation in Python Using SKLearn**. Disponível em: <https://www.askpython.com/python/examples/k-fold-cross-validation>, 2020. Acesso em: 13 de novembro de 2023.

YANG, J.; YANG, C.; HU, X. A study of hybrid recommendation algorithm based on user. In: IEEE. **2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)**. [S.l.], 2016. v. 2, p. 261–264.

ZHANG, H. The optimality of naive bayes. **Aa**, v. 1, n. 2, p. 3, 2004.

ZHU, Y. Personalized recommendation of educational resource information based on adaptive genetic algorithm. **International Journal of Reliability, Quality and Safety Engineering**, World Scientific, v. 30, n. 02, p. 2250014, 2023.