

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE SISTEMAS DE INFORMAÇÃO

ISMAEL AGUIRRE LEAL

**UM MÓDULO DE RECOMENDAÇÃO DE PROJETOS BASEADO EM MACHINE
LEARNING PARA O SISTEMA DE GESTÃO DE TCC DO INE**

Florianópolis

2022

Ismael Aguirre Leal

**UM MÓDULO DE RECOMENDAÇÃO DE PROJETOS BASEADO EM MACHINE
LEARNING PARA O SISTEMA DE GESTÃO DE TCCS DO INE**

Trabalho de Conclusão de Curso apresentado
como parte dos requisitos para a obtenção do
grau de Bacharel em Sistemas de Informação
pela Universidade Federal de Santa Catarina.

Orientador: Prof. Dr. Jean Carlo Rossa Hauck

Florianópolis

2022

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC

Leal, Ismael
UM MÓDULO DE RECOMENDAÇÃO DE PROJETOS BASEADO
EM MACHINE
LEARNING PARA O SISTEMA DE GESTÃO DE TCCS DO INE
/ Ismael Leal ; orientador, Jean Carlo Rossa
Hauck, 2022.
106 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro
Tecnológico, Graduação em Sistema de Informação,
Florianópolis, 2022.

Inclui referências.

1. Sistema de Informação. 2. Sistema de
recomendação. 3. Machine Learning. I. Carlo
Rossa Hauck, Jean. II.
Universidade Federal de Santa Catarina.
Graduação em Sistema de Informação. III. Título.

Ismael Aguirre Leal

**UM MÓDULO DE RECOMENDAÇÃO DE PROJETOS BASEADO EM MACHINE
LEARNING PARA O SISTEMA DE GESTÃO DE TCCS DO INE**

Este Trabalho Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Sistema de Informação e aprovado em sua forma final pelo Curso Sistema de Informação.

Florianópolis, 22 de dezembro de 2022.

Prof. Álvaro Junio Pereira Franco, Dr.
Coordenador do Curso

Banca Examinadora:

Prof. Dr. Jean Carlo Rossa Hauck
Orientador
Universidade Federal de Santa Catarina

Prof. Dr. Renato Cislighi
Banca Examinadora
Universidade Federal de Santa Catarina

Prof. Dr. Elder Rizzon Santos
Banca Examinadora
Universidade Federal de Santa Catarina

Este trabalho é dedicado, primeiramente, a minha família, que sempre apoiou minhas decisões e esteve ao meu lado em todas as horas. Em especial, dedico este trabalho a minha mãe, que detém metade do diploma que virá e ao meu querido pai, que detém a outra metade e, infelizmente, não estará aqui para receber a sua metade. Também o dedico a minha noiva, que além de sempre me apoiar, conseguiu me suportar durante o desenvolvimento deste trabalho. Eu não estaria me formando ou sequer estaria nesse curso sem a ajuda dela. Não podendo ficar de fora, agradeço aos meus queridos colegas, que se tornaram grandes amigos, por toda parceria durante o curso. O curso não seria o mesmo sem vocês. Homenageio ainda os meus amados irmãos, que sempre foram exemplos, cada um à sua maneira, e nortearam meu caminho até aqui. Entre os irmãos, homenagem especial para a Camila, exemplo de força e perseverança para todos.

RESUMO

Os alunos dos cursos de Sistemas de informação ou Ciências da Computação da Universidade Federal de Santa Catarina realizam um Trabalho de Conclusão de Curso (TCC) obrigatório como requisito para a obtenção do diploma. Atualmente o Sistema de Gestão de TCCs do INE/CTC/UFSC proporciona ao aluno de Sistemas de informação ou Ciências da Computação uma busca rápida de possíveis orientadores e seus respectivos interesses em áreas e temas de pesquisa. Apesar dessa possibilidade de busca por parte dos alunos, geralmente os temas propostos são apresentados de forma genérica e abrangente, não permitindo ao aluno saber exatamente o que poderia ser desenvolvido no seu TCC. Alguns professores fazem uso de outras ferramentas, como vídeos e links, explicando e demonstrando os temas de forma mais específica. Por sua vez, para o aluno, há pouco suporte da ferramenta de gestão de TCCs na escolha de um tema para o trabalho. A apresentação mais detalhada de temas por parte dos professores, bem como a possibilidade de uma recomendação de tema ou área, para os alunos, de acordo com as possíveis aptidões e afinidades apresentadas durante o curso de graduação, poderiam facilitar a escolha do tema do TCC e possivelmente melhorar o resultado ao final do TCC. Assim, este trabalho apresenta o desenvolvimento de um módulo de recomendação de projetos baseado em *Machine Learning* para o sistema de gerenciamento de TCCs do INE/CTC/UFSC. Um dataset é desenvolvido a partir dos dados de dois diferentes sistemas, o CAGR e o Sistema de Gestão de TCCs do INE, sendo extraídos através de query e logs e organizado manualmente, de acordo com a necessidade. O modelo de Machine Learning é implementado utilizando linguagem de código Python em um Jupyter Notebook no ambiente do Google Colab. O módulo é modelado e desenvolvido utilizando, entre outras, a biblioteca FastAi, que facilita a implementação do modelo e integrado ao sistema de TCCs. Uma avaliação inicial do módulo desenvolvido levantou indícios de que o novo módulo é muito útil para os alunos, além de ter uma boa usabilidade e funcionalidade, sendo fácil de usar.

Palavras-chave: inteligência artificial, recomendação, aprendizagem de máquina, TCC, sistema web.

ABSTRACT

Currently the INE / CTC / UFSC CCT management system provides the student of Information Systems and Computer Science with a quick check of possible advisors and their respective interests in research areas and topics. Despite this possibility of checking interests, on the part of the students, generally the themes are of a generic and very comprehensive nature, not allowing the student to know exactly what could be developed in a CBT. Some teachers make use of other tools, such as videos on the Youtube platform, explaining and demonstrating the topics in a more specific way, clarifying some of the student's doubts. In turn, for the student, there is no help in choosing a topic for the job, especially when there are only lists of generic topics. An alternative would be to make it possible for teachers to present more specific themes, with the theme, research area and more details about a possible job, as well as the possibility of a recommendation of a theme or area, for students, according to the skills presented during the undergraduate course. Thus, this proposal presents the development of a project recommendation module based on Machine Learning for the INE / CTC / UFSC TCC management system. A dataset is developed from data from two different systems, CAGR and INE's TCCs Management System, being extracted through queries and logs and organized manually, according to need. The Machine Learning model is implemented using the Python code language in a Jupyter Notebook in the Google Colab environment. An initial evaluation of the module developed with signs that the new module is very useful for students, in addition to having good usability and functionality, being easy to use.

Keywords: artificial intelligence, recommendation, machine learning, CBT, web system.

LISTA DE FIGURAS

Figura 1 - Diagrama de Euler para Machine Learning (GAVRILOVA, 2020).	27
Figura 2 - Fluxo ML (THAKUR, 2020).....	28
Figura 3 - - Áreas de aprendizagem de ML (PATEL, 2018).	29
Figura 4 - Aprendizado supervisionado (ESCOVEDO, 2020).	30
Figura 5 - Aprendizado não supervisionado (ESCOVEDO, 2020).....	31
Figura 6 - Aprendizado por reforço (TECHVIDVAN, 2020).....	32
Figura 7 - The neural network zoo (VAN VEEN, 2019)	34
Figura 8 - Aprendizado de máquina vs. aprendizado profundo para reconhecimento facial (ROBINS, 2020).....	35
Figura 9 - Filtragem colaborativa e filtragem baseada em conteúdo (MORANDINO; RODRIGUES, 2020).....	37
Figura 10 - Representação da ordem de recomendação dos itens para o usuário "A", LEON (2021).....	38
Figura 11 - Recomendação em modelo de grafo, SALLES E WILLRICH (2015) ...	38
Figura 12 - Organizações de ontologias, Brascher e Café (2008).....	39
Figura 13 - Protótipo US01	52
Figura 14 - Protótipo US02	53
Figura 15 - Protótipo US03	54
Figura 16 - Protótipo US04	55
Figura 17 - Fluxograma de configuração de ambiente de desenvolvimento.....	58
Figura 18 - Organização do projeto e arquivo pom.xml.	59
Figura 19 - Organização da base de dados e exemplo de tabela de dados.	60
Figura 20 - Configuração utilizada em ambiente de desenvolvimento	62
Figura 21 - Recorte do <i>dataset</i> com dados estruturados	65
Figura 22 - Recorte do <i>dataset</i> importado no código Python.....	66
Figura 23 - Trecho do código com função de recomendação	66
Figura 24 - Trecho de código com modelo usando SVD	67
Figura 25 - Arquitetura de desenvolvimento do modelo de recomendação.....	68
Figura 26 - Trecho do código com separação de variáveis e variável alvo.....	68
Figura 27 - Trecho do código com divisão de <i>dataset</i>	69

Figura 28 - Trecho do código com criação do modelo e gráfico de perda x aprendizado	69
Figura 29 - Trecho de código com recorde das épocas do treinamento do modelo ...	70
Figura 30 - Trecho do código com recorde de predição realizada	71
Figura 31 - Trecho do código com valores de validação e exportação do modelo	71
Figura 32 - Arquitetura do serviço de recomendação	72
Figura 33 - Trecho do código que realiza predição do <i>dataset</i> de teste	73
Figura 34 – Ilustração mostrando na coluna 1 o número de épocas, na coluna 2 a perda e na coluna 4 a eficácia.....	74
Figura 35 - Planilha com dados conhecidos.....	74
Figura 36 - Planilha conhecida com dados excluídos	75
Figura 37 - Planilha com dados excluídos importada no código do modelo	75
Figura 38 - Recorte de tabela usada para validação da recomendação	75
Figura 39 - Resultado da recomendação para cada caso conhecido de teste	76

LISTA DE TABELAS

Tabela 1 - Termos de busca.....	43
Tabela 2 - Resultado da busca.....	44
Tabela 3 - Requisitos funcionais	50
Tabela 4 - Tabela de áreas para classificação de TCCs	63
Tabela 5 - Tabela com disciplinas da área de Tratamento e Banco de Dados	64
Tabela 6 - Questionário de satisfação.....	81
Tabela 7 - Respostas sobre usabilidade	83
Tabela 8 - Respostas sobre desempenho	83
Tabela 9 - Respostas sobre funcionalidade	84
Tabela 10 - Pontuação SUS.....	85

LISTA DE QUADROS

Quadros 1 - Design de interface US01	77
Quadros 2 - Design de interface US02	77
Quadros 3 - Design de interface US03	78
Quadros 4 - Design de interface US04	78

LISTA DE ABREVIATURAS E SIGLAS

TCC - Trabalho de conclusão de curso

INE - Departamento de informática e estatística

UFSC - Universidade Federal de Santa Catarina

CAGR - Sistema de Controle Acadêmico da Graduação

SIN - Sistemas de Informação

CCO - Ciências da computação

WPA - Aplicativo Web Progressivo

SOAP - Simple Object Access Protocol ou Protocolo Simples de Acesso a Objetos

JSF - JavaServer Faces

JSP - JavaServer Pages

XML - Extensible Markup Language ou

API - Application Programming Interfaces ou

W3C - World Wide Web Consortium

ACID - Atomicidade, Consistência, Isolamento e Durabilidade

HTTP - Hypertext Transfer Protocol ou Protocolo de Transferência de Hipertexto

SMTP - Simple Mail Transfer Protocol ou Protocolo de Transferência de Correio Simples

TPC - Transmission Control Protocol ou Protocolo de Controle de Transmissão

JVM - Java Virtual Machine ou Máquina Virtual Java

IoC - Inversion of Control ou Inversão de Controle

DI - Dependency Injection ou Injeção de Dependências

AOP - Aspect Oriented Programming ou Programação orientada a aspecto

SGBD - Data Base Management System ou Sistema Gerenciador de Banco de Dados

SQL - Structured Query Language ou Linguagem de Consulta Estruturada

JDBC - Java Database Connectivity

IA - Inteligencia Artificial

ML - Machine Learning ou Aprendizado de Máquina

NN - Neural Nets ou Redes Neurais

DL - Deep Learning ou aprendizado profundo

AM - Aprendizado de Máquina

CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior

EaD - Ensino à Distância

IDE - Ambiente de desenvolvimento integrado

SETIC - Superintendência de Governança Eletrônica e Tecnologia da Informação e Comunicação

SUMÁRIO

1	INTRODUÇÃO	15
1.1	OBJETIVOS	15
1.1.1	Objetivo Geral.....	16
1.1.2	Objetivos Específicos	16
1.2	MÉTODO DE PESQUISA.....	17
1.3	ESTRUTURA DO TRABALHO	19
2	FUNDAMENTAÇÃO TEÓRICA.....	20
2.1	SISTEMA DE GESTÃO DE TCC DO INE	20
2.2	WEB SERVICES SOAP	21
2.3	JAVASERVER FACES	23
2.4	SPRING	23
2.5	MYSQL	24
2.6	HIBERNATE.....	25
2.7	<i>MACHINE LEARNING</i>	26
2.7.1	Aprendizagem supervisionada	29
2.7.2	Aprendizagem não supervisionada	30
2.7.3	Aprendizado de máquina reforçado	31
2.7.4	Redes Neurais.....	32
2.7.5	Deep Learning.....	35
2.8	SISTEMAS DE RECOMENDAÇÃO	36
2.8.1	Sistemas de recomendação abordagem clássica.....	36
2.8.2	Sistemas de recomendação baseado em grafos	37
2.8.3	Sistemas de recomendação baseado em ontologia	39
2.9	BIBLIOTECA FASTAI	40

2.10	GOOGLE COLAB E JUPYTER NOTEBOOK.....	41
3	TRABALHOS CORRELATOS	43
3.1	DEFINIÇÃO DO PROTOCOLO DE PESQUISA	43
3.1.1	Critérios de Inclusão e Exclusão:	44
3.2	EXTRAÇÃO DE INFORMAÇÃO	44
3.3	EXTRAÇÃO DE INFORMAÇÃO	45
3.3.1	Recomendação Assistida por Computador para Orientação de Trabalhos de Conclusão de Curso: Baseada no Perfil Informacional e na Capacidade de Orientação Docente 45	
3.3.2	Uma introdução aos sistemas de recomendação, modelos matemáticos, algoritmos e aplicações.....	46
3.3.3	Interpretabilidade em Modelos de Sistemas de Recomendação.....	47
3.3.4	Considerações Finais	48
4	DESENVOLVIMENTO.....	50
4.1	ANÁLISE DE REQUISITOS	50
4.2	IMPLEMENTAÇÃO.....	56
4.2.1	Seleção de tecnologias.....	56
4.2.2	Preparação do ambiente.....	57
4.2.3	Preparação do conjunto de dados	62
4.2.4	Treinamento do modelo	65
4.2.5	Implementação de modelo de recomendação.....	72
4.2.6	Avaliação do modelo de recomendação	73
4.2.7	Design de interface do módulo de recomendação.....	76
5	Avaliação do módulo de recomendação.....	80
5.1	AVALIAÇÃO POR USUÁRIOS.....	80
5.1.1	Definição da avaliação por usuários	80
5.1.2	Execução da avaliação por usuários	82
5.1.3	Análise dos dados.....	82

6	Conclusão	86
	REFERÊNCIAS	89

1 INTRODUÇÃO

É muito comum que alunos de cursos de graduação cheguem às últimas fases dos seus cursos sem ideia alguma sobre qual tema escolher para o seu Trabalho de Conclusão de Curso (TCC). A escolha de um objetivo de pesquisa é frequentemente a tarefa mais difícil em um trabalho de mestrado ou doutorado (WAZLAWICK, 2009), o que também costuma valer para um trabalho de conclusão de curso. Então, alunos iniciando seus TCCs acabam percebendo que possuem pouco tempo para buscar um tema de seu interesse e que combine com um orientador de sua afinidade. Ainda segundo Wazlawick (2009), não se recomenda uma pesquisa cujo tema não seja compatível com os conhecimentos do orientador. Somente depois da escolha do tema, do objetivo e do orientador é que o aluno começará o seu trabalho (SIN, 2011).

O Departamento de Informática e Estatística (INE) da Universidade Federal de Santa Catarina (UFSC), que oferece os cursos de Sistemas de Informação (SIN) e Ciências da Computação (CCO), possui um *software*¹ para gerenciamento de TCCs do departamento. Trata-se de um Sistema de Gestão de Trabalhos de Conclusão de Curso que permite a entrega de documentos e acompanhamento de calendário das disciplinas vinculadas ao TCC, por parte do aluno. Também permite aos orientadores e membros de bancas acompanharem e avaliarem os projetos (BOTELHO; UGIONI, 2015).

Atualmente, o Sistema de Gestão de TCC do INE permite visualizar uma lista de professores e suas respectivas áreas de pesquisa, nas quais se interessam por orientar trabalhos. Porém o sistema carece de um ambiente que fomente ideias de temas de pesquisa mais específicos para os alunos. Segundo Reis (2008), o tema a ser escolhido depende de vários aspectos, como o interesse do aluno, a experiência pessoal e, portanto, para tal escolha é vital ao aluno priorizar o assunto e refletir sobre a escolha a ser feita. Quanto maior for o leque de ideias e temas, maior tende a ser a chance de o aluno encontrar algo que seja de seu interesse. Assim, o desempenho dos alunos nas diversas disciplinas, no decorrer do curso, poderia ser utilizado para indicar potenciais temas de interesse.

1.1 OBJETIVOS

¹ <https://tcc.inf.ufsc.br/>

Nas seções a seguir estão descritos o objetivo geral e os objetivos específicos deste TCC.

1.1.1 Objetivo Geral

O presente trabalho tem como objetivo desenvolver um módulo de recomendação de temas de projetos baseado em *Machine Learning* para o Sistema de Gerenciamento de TCCs do INE. No módulo de recomendação serão apresentadas as áreas e temas nas quais professores têm interesse em orientar, bem como um modelo de inteligência artificial que será treinado com os dados passados de desempenho de alunos nas disciplinas e o sucesso nos TCCs, para recomendar áreas ou temas de possível interesse do aluno para o seu trabalho de conclusão de curso. Os dados serão obtidos diretamente do Sistema de Controle Acadêmico de Graduação² (CAGR) dos cursos de SIN e CCO.

O novo módulo trará ‘*cards*’ com temas e projetos, previamente definidos pelos professores, para eventual interesse dos alunos. Também possibilitará a recomendação de temas de TCCs aos alunos, utilizando seus dados de histórico escolar disponíveis no CAGR para trazer temas, projetos ou áreas que possam ser de seu interesse. Essa recomendação será feita por um modelo de inteligência artificial que usará dados de alunos de Sistemas de Informação e Ciências da Computação da UFSC, disponíveis no CAGR e, se necessário, outros sistemas pertinentes, para ser treinada.

1.1.2 Objetivos Específicos

- O1. Análise do estado da arte em sistemas de recomendação
- O2. Levantamento e análise dos requisitos para um módulo de recomendação de temas de projetos de TCC
- O3. Treinamento de um modelo de Inteligência Artificial (IA) para recomendação de temas de TCC

² <https://cagr.sistemas.ufsc.br/>

O4. Implementação de um módulo de recomendação de temas de TCC para o sistema Gerenciamento de TCCs do INE

O5. Avaliação do novo módulo

1.2 MÉTODO DE PESQUISA

A metodologia de pesquisa utilizada, visando alcançar os objetivos do presente trabalho, é dividida nas seguintes etapas:

Etapa 1: Fundamentação teórica

Nesta etapa será realizado o estudo e análise de leitura sobre os principais conceitos e teorias a serem abordados ao longo do trabalho. A etapa é composta pelas seguintes atividades:

Atividade 1.1: Análise da literatura e conceitos envolvidos na realização de trabalhos de conclusão de curso no INE.

Atividade 1.2: Análise da literatura e conceitos envolvidos nas tecnologias empregadas no desenvolvimento do Sistema de Gerenciamento de TCCs do INE.

Atividade 1.3: Análise da literatura e conceitos de *Machine Learning* (ML) e sistemas de recomendação.

Etapa 2: Estado da arte

Nesta etapa será realizada o levantamento e análise do estado da arte utilizando a técnica, proposta por Kitchenham (2004) de mapeamento sistemático da literatura analisada.

Atividade 2.1: Definição de protocolo da revisão.

Atividade 2.2: Busca e seleção de trabalhos relevantes.

Atividade 2.3: Análise de informações dos trabalhos relevantes.

Etapa 3: Engenharia de requisitos

Esta etapa contempla a engenharia de requisitos com coleta e análise de dados para entender as necessidades e apresentar possíveis melhorias de usuários nos perfis de professor e de aluno.

Atividade 3.1: Elaboração e realização de questionário com professores do INE para identificação de possíveis problemas e coleta de dados.

Atividade 3.2: Análise de resultados obtidos no questionário.

Atividade 3.3: Criação de requisitos.

Etapa 4: Desenvolvimento do módulo

Nesta etapa, será implementado o sistema de recomendação com inteligência artificial, junto com as modificações necessárias no Sistema de Gestão de TCC do INE para contemplar o novo módulo.

Atividade 4.1: Análise de requisitos.

Atividade 4.2: Prototipação de telas.

Atividade 4.3: Configuração de ambientes de desenvolvimento.

Atividade 4.4: Implementação de módulo proposto.

Etapa 5: Desenvolvimento do modelo de IA

Nesta etapa será desenvolvido o modelo de inteligência artificial que fará a recomendação de temas e áreas no novo módulo do Sistema de Gestão de TCC do INE.

Atividade 5.1: Coleta e preparação dos dados.

Atividade 5.2: Escolha do modelo.

Atividade 5.3: Treinamento do modelo.

Atividade 5.4: Refinamento do modelo.

Atividade 5.5: Avaliação do modelo.

Atividade 5.6: Aprimoramento dos parâmetros

Atividade 5.7: Predição

Etapa 6: Avaliação de resultados

Nesta etapa, será avaliado a qualidade da inteligência artificial do sistema de recomendação, bem como a usabilidade e funcionalidade das modificações realizadas no presente trabalho.

Atividade 6.1: Avaliação da inteligência artificial.

Atividade 6.2: Teste do sistema

1.3 ESTRUTURA DO TRABALHO

O que vem a seguir neste trabalho estará estruturado da seguinte maneira. O capítulo 2 traz as informações e conceitos que estão fundamentando este trabalho. O capítulo 3 apresenta os trabalhos correlatos. O capítulo 4 aborda uma análise de requisitos que foram levantados para implementação da solução proposta, bem como os passos para preparação de ambiente de desenvolvimento. O capítulo 5 traz as avaliações feitas para o módulo de recomendação desenvolvido e a análise dos dados obtidos através das avaliações. Já no capítulo 6 é apresentada a conclusão do trabalho, passando pelos pontos fortes, pontos de melhoria e das possibilidades para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados alguns conhecimentos e conceitos pertinentes para a compreensão do trabalho. Além de uma ideia de conceitos do Sistemas de TCCs do INE, serão abordadas as tecnologias utilizadas no sistema, a fim de gerar um melhor entendimento.

2.1 SISTEMA DE GESTÃO DE TCC DO INE

O INE, Departamento de Informática e Estatística da Universidade Federal de Santa Catarina, conta com um sistema de gerenciamento de projetos de conclusão de curso (TCCs), sendo utilizado tanto por alunos quanto por professores para realizar as mais variadas atividades relacionadas à gestão de TCCs. O sistema foi concebido e iniciado por meio de um trabalho de conclusão de curso, onde os alunos desenvolveram o sistema (BOTELHO; UGIONI, 2013).

O Departamento de Informática e Estatística da Universidade Federal de Santa Catarina, até a conclusão do sistema proposto por Botelho e Ugioni (2013), utilizava outro sistema para gestão dos TCCs, desenvolvido por Lanzarin (2004). Esse sistema antigo não possuía integração com os demais sistemas que a UFSC utiliza, deixando-o restrito à automação de algumas funções e tarefas, como a exibição de notas lançadas no Sistema de Controle Acadêmico da Graduação (CAGR), bem como à vinculação de cadastro de alunos matriculados em disciplinas do contexto de trabalhos de conclusão de curso.

O Sistema de Gestão de TCC do INE vem sendo desenvolvido quase que de forma colaborativa, tendo funcionalidades incorporadas e melhorias realizadas por alunos em seus trabalhos de conclusão de curso. Um desses trabalhos, (GONÇALVES, 2016) elaborou melhorias significativas na usabilidade do sistema, focando principalmente nos usuários com perfil de aluno. Outros dois trabalhos trouxeram melhorias ao sistema, tendo o aluno (ONGHERO, 2018) realizado melhorias de funcionalidades, como permitir a definição de área de conhecimento, permitir a visualização de TCC em que é membro, visualização de membro

e avaliações, além de melhorias de desempenho. O outro trabalho realizado em trabalho de conclusão de curso (BAPTISTA, 2018), propôs o desenvolvimento do sistema em uma aplicação WPA (Aplicativo Web Progressivo) para permitir uma melhor usabilidade e um ganho de desempenho quando o sistema for acessado em dispositivos móveis.

O Sistema de Gestão de TCC do INE vem sendo desenvolvido sobre uma plataforma web SOAP (*Simple Object Access Protocol*, em português Protocolo Simples de Acesso a Objetos), tendo o auxílio de um *framework* de desenvolvimento de aplicações próprio da UFSC, que é utilizado em sistemas internos. O sistema conta com a linguagem de programação Java³ e tecnologias e ferramentas como os *frameworks* de aplicações Spring⁴ e JSF⁵ (JavaServer Faces), o Hibernate⁶ como *framework* de mapeamento objeto-relacional para a persistência de dados. Também é utilizado o MySQL⁷ como gerenciado de banco de dados. A seguir, neste mesmo capítulo do trabalho, tais tecnologias serão mais bem detalhadas.

2.2 WEB SERVICES SOAP

Os *Web Services* são apresentados como um meio de integração entre sistemas, independentemente da plataforma de *hardware* e de *software*, onde, em geral, a troca de informações entre dispositivos, principalmente dispositivos móveis, é realizada com web services (GOMES, 2014). SOAP, sigla utilizada para *Simple Object Access Protocol*, é um formato muito comum de protocolo de empacotamento para as mensagens compartilhadas por aplicativos, baseado em XML⁸ (*Extensible Markup Language*) (Kulchenko, Snell and Tidwell, 2001). O XML é uma linguagem de marcação que pode ser lida por humanos e por máquina. O SOAP define o que é chamado comumente de APIs (*Application programming interfaces*), que proporcionam flexibilização, simplicidade, segurança e controle, na conexão entre diferentes programas e *softwares*. Estes mesmos autores, (KULCHENKO, SNELL AND TIDWELL, 2001), também afirmam que a especificação define nada mais do que um envelope simples para

³ <https://www.java.com/pt-BR/about/>

⁴ <https://spring.io/>

⁵ <https://javaee.github.io/javaxserverfaces-spec/>

⁶ <https://hibernate.org/>

⁷ <https://www.mysql.com/>

⁸ <https://www.w3.org/TR/REC-xml/>

as informações que estão sendo transferidas e um conjunto de regras para converter aplicativos e tipos de dados específicos da plataforma em representações XML. O protocolo de acesso a objetos simples (SOAP) é um protocolo oficial mantido pelo *World Wide Web Consortium* (W3C).

Segundo a boa descrição da gigante de *Softwares* RedHat⁹, em seu site, o SOAP, por ser um protocolo, impõe regras integradas que podem aumentar a complexidade e sobrecarga das aplicações, o que pode levar a tempos de carregamento de página mais longos. No entanto, esses padrões também oferecem conformidades integradas que podem torná-los preferíveis para cenários corporativos. Os padrões de conformidade integrados incluem segurança, atômidade, consistência, isolamento e durabilidade (ACID), que é um conjunto de propriedades para garantir transações de banco de dados confiáveis.

As especificações comuns de serviços da web incluem:

- **Segurança de serviços da Web (WS-security):** padroniza como as mensagens são protegidas e transferidas por meio de identificadores exclusivos chamados *tokens*.
- **WS-ReliableMessaging:** Padroniza o tratamento de erros entre mensagens transferidas em uma infraestrutura de TI não confiável.
- **Endereçamento de serviços da Web (WS-addressing):** Pacotes de informações de roteamento como metadados dentro de cabeçalhos SOAP, em vez de manter essas informações mais profundas na rede.
- **Linguagem de descrição de serviços da Web (WSDL):** Descreve o que um serviço da Web faz e onde esse serviço começa e termina.

Quando uma solicitação de dados é enviada a uma API SOAP, ela pode ser tratada por meio de qualquer um dos protocolos da camada de aplicativo: HTTP (para navegadores), SMTP (para e-mail), TCP e outros. No entanto, uma vez que uma solicitação é recebida, as mensagens SOAP de retorno devem ser retornadas como documentos XML.

⁹ <https://www.redhat.com/en/about/company>

2.3 JAVASERVER FACES

O JavaServer Faces (JSF) é um *framework* de interface de usuários para aplicações web, baseado em *servlets*, que foi desenvolvido através do *Java Community Process*. De acordo com o conteúdo da página da própria¹⁰ tecnologia, o JSF foi projetado para ser flexível e potencializar os conceitos de interface de usuário existentes, sem limitar os desenvolvedores a uma linguagem de marcação, protocolos ou dispositivos clientes específicos.

Segundo Geary e Hostmann (2010), Java Server Faces promete disponibilizar o rápido desenvolvimento de interfaces de usuário para o lado servidor Java, permitindo aos desenvolvedores escrever sem dificuldades aplicações, sem se preocupar com as complexidades de lidar com navegadores e servidores Web. Também automatiza detalhes tediosos de baixo nível, tais como o fluxo de controle e o transporte do código entre formulários Web e a lógica de negócios. JSF oferece muita flexibilidade em como você realmente desenvolve a interface do usuário. Todas as implementações JSF são necessárias para suportar JavaServer Pages (JSP) como uma tecnologia de camada de apresentação, com componentes JSF representados por elementos de ação customizada. A API JSF, no entanto, é flexível o suficiente para suportar outras tecnologias de apresentação além do JSP. Por exemplo, você pode usar código Java puro para criar componentes JSF (BERGSTEN, 2009).

2.4 SPRING

Spring Framework é um *framework open-source* de aplicações para a plataforma Java, que faz uso da JVM (Java Virtual Machine) e é baseado nos conceitos de inversão de controle (IoC) e injeção de dependências (DI) (WEISSMANN, 2014). É um *framework* com o intuito de tornar as aplicações mais simples, ágeis e flexíveis (SPRING, 2017). O Spring começou a ser desenvolvido em 2003, depois de Rod Johnson, em um marco histórico para o desenvolvimento de aplicações corporativas (WEISSMANN, 2014), publicar seu livro *One-on-One: J2EE Design and Development* (JOHNSON, 2002), porém ele só foi apresentado oficialmente em 2004 com o livro *Expert One-to-One J2EE Development Without EJB* (JOHNSON, 2004).

¹⁰ <https://www.oracle.com/java/technologies/javaserverfaces.html>

O autor de *Spring in Action* (2015) e membro da equipe de engenharia da Spring, Craig Walls, define o *framework* como eficiente, porém leve e que não exige o uso de EJBs, reduzindo significativamente a complexidade do uso de interfaces, agilizando e simplificando o desenvolvimento de uma aplicação. Isso se dá pela utilização dos conceitos: Injeção de dependências, que é uma especialização de Inversão de Controle, container Spring e programação orientada à aspectos (AOP - *Aspect Oriented Programming*). Ainda, segundo Martin Fowler (2004), inversão de controle é uma característica comum dos *frameworks*, portanto, não devemos afirmar que os *containers* são leves e especiais por usarem esse conceito mais genérico, e sim por usarem o que foi estabelecido com o nome de *Dependency Injection* (injeção de dependência) que consiste em ter um objeto separado para popular um campo em um objeto, com uma implementação própria para uma interface. De um modo mais objetivo, isso reduz o acoplamento entre os componentes da aplicação. Já a programação orientada a aspectos facilita o tratamento dos problemas dos acoplamentos relacionados a funcionalidades verificados na aplicação (WEISSMANN, 2014)

2.5 MYSQL

O MySQL é um Sistema Gerenciador de Banco de Dados (SGBD) ou, em inglês, *Data Base Management System* (DBMS) muito popular, sendo o segundo colocado no ranking criado pelo site especializado Db-engines¹¹ devido as suas qualidades como simplicidade, desempenho e estabilidade, sendo utilizado, segundo a própria MySQL, por grandes empresas tais como Facebook, Google e Adobe, entre outras. O MySQL é um SGBD *open-source* que foi testado e documentado de forma mais completa do que outros sistemas de banco de dados, o que torna relativamente fácil encontrar desenvolvedores com experiência em MySQL (KOFLER, 2004). As raízes do MySQL começaram em meados dos anos oitenta, porém só na versão 3.11.1, lançada em 1996, foi contemplada com versão binária para os sistemas Linux e Solaris. A empresa MySQL AB foi formada para oferecer a distribuição de MySQL e ofertar serviços

¹¹ db-engines.com

comerciais. Em 2008, a Sun Microsystems adquiriu a MySQL AB e, em 2010, a Oracle adquiriu a Sun. Hoje, o MySQL está disponível em formato binário e fonte e funciona em muitas outras plataformas. (DUBOIS, 2013).

2.6 HIBERNATE

O Hibernate é um *framework open-source* para o mapeamento objeto-relacional (ORM do inglês *Object/Relational Mapping*), escrito em Java e também disponível em .NET¹². De um modo geral, o mapeamento objeto-relacional é a persistência automatizada e transparente de objetos em uma aplicação Java para as tabelas em um banco de dados SQL (*Structured Query Language* ou Linguagem de Consulta Estruturada), usando metadados que descrevem esse mapeamento entre as classes da aplicação e o esquema do banco de dados SQL (GREGORY and BAUER, 2015). Ainda por Gregory e Bauer (2015), uma suposta vantagem do ORM é que ele protege os desenvolvedores Java de SQL confuso. Eles acreditam que os desenvolvedores devem ter um nível suficiente de familiaridade de modelagem relacional e SQL para trabalhar com o Hibernate. Para efetivamente usar o Hibernate você deve ser capaz de ver e interpretar as instruções SQL que ele emite e entender suas implicações de desempenho.

A maioria das pessoas começa a se esforçar para escrever algumas consultas SQL, incorporando-as desajeitadamente no código-fonte, como *strings*, e trabalhando com conectividade de banco de dados Java (JDBC) para executá-las e processar os resultados. O JDBC evoluiu para uma biblioteca de comunicação de banco de dados rica e flexível, que agora fornece maneiras de simplificar e melhorar essa abordagem, mas ainda há um certo grau de tédio envolvido. Houve então a necessidade de mover as consultas para fora do código, fazendo agirem mais como componentes (O'BRIEN et al., 2008). Essa nova maneira de manipulação e operação de aplicações e banco de dados baixou a complexidade, aumentando a eficiência no desenvolvimento das aplicações. Segundo O'Brien et al. o desenvolvimento do Hibernate foi claramente um divisor de águas na interação entre Java e bancos de dados relacionais.

¹² <https://dotnet.microsoft.com/>

2.7 MACHINE LEARNING

A Inteligência Artificial (IA) é um grande campo que abrange lógica, probabilidade e matemática, além de percepção, raciocínio, aprendizado, ação e, ainda, tudo o que se refere à eletrônica, desde dispositivos microeletrônicos até robôs (RUSSEL & NORVIG, 2013). O termo 'Inteligência Artificial' foi cunhado por John McCarthy, um cientista da computação, em 1955. Os mesmos autores, Russel & Norvig (2013), definem sucintamente a inteligência artificial como o estudo de agentes inteligentes que recebem percepções e dados do ambiente e executam ações (RUSSEL & NORVIG, 2013). Um agente inteligente aprenderá caso melhore o seu desempenho em alguma tarefa futura, após fazer observações sobre o mundo. Nesse sentido, pode-se dizer que se trata de um aprendizado de máquina.

Com o avanço da tecnologia, surgiram diversos campos de estudo e subconjuntos na Inteligência Artificial. Dentre as diversas áreas desse vasto campo, o presente trabalho irá se ater na parte de *Machine Learning*. Figura 1 mostra um diagrama de Euler¹³ para IA e seus derivados.

¹³ https://pt.wikipedia.org/wiki/Diagrama_de_Euler

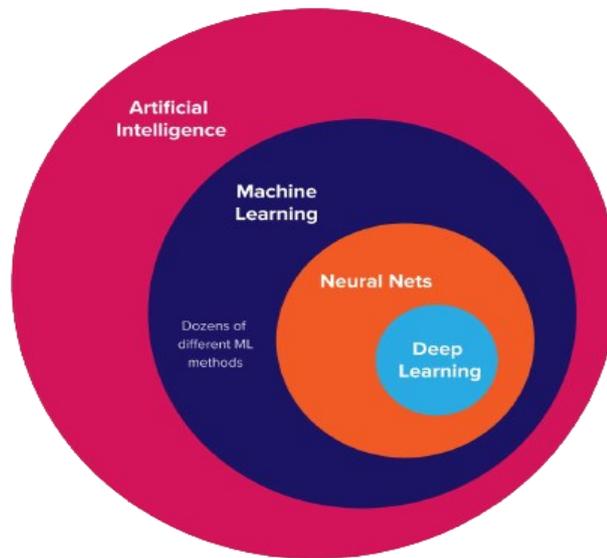


Figura 1 - Diagrama de Euler para Machine Learning (GAVRILOVA, 2020).

O *Machine Learning* (ML), também conhecido como Aprendizado de Máquina (AM), é um subconjunto da Inteligência Artificial que explora o estudo e a construção de algoritmos que podem aprender com os dados. Arthur Samuel, outro cientista da computação, define, em 1959, *Machine Learning* como “o campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados” (SAMUEL, 1959). Já outro autor define ML como um ramo da IA que estuda programas que aprendem com base em experiência a desenvolver melhor uma tarefa, segundo uma medida de desempenho. Ou seja, possuem a capacidade de melhorar o desempenho na realização de alguma tarefa por meio da experiência (MITCHEL, 1997). Neste sentido a Figura 2 ilustra, de modo simplificado, o modo de funcionamento de um ML, que possui a capacidade de aprender com um treinamento onde são fornecidos um grande volume de dados e os resultados. Assim, o algoritmo consegue aplicar o conhecimento adquirido em um novo conjunto de dados.

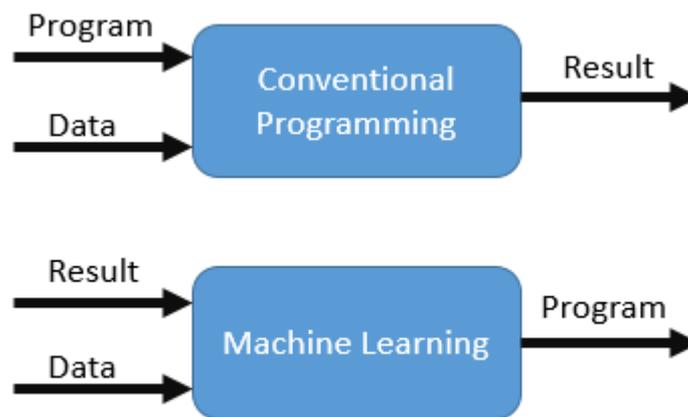


Figura 2 - Fluxo ML (THAKUR, 2020).

A capacidade de se adaptar, melhorar o seu comportamento e suas respostas é o principal objetivo em ML. Se diz estar aprendendo (treinando, construindo, formulando) a partir de um conjunto de dados, quando se procura padrões no conjunto fornecido ao algoritmo (ESCOVEDO; KOSHIYAMA, 2020). Podemos dividir ML em dois grandes grupos de aprendizado: aprendizado supervisionado e aprendizado não supervisionado. Ainda temos um terceiro grande grupo, que vale ser mencionado, o aprendizado por esforço (SUTTON; BARTO, 1998). Os modelos de aprendizados supervisionados e não supervisionados também são chamados, respectivamente, de modelos preditivos e modelos descritivos (FACELI et al., 2015). A Figura 3 identifica a diversidade de modelos.

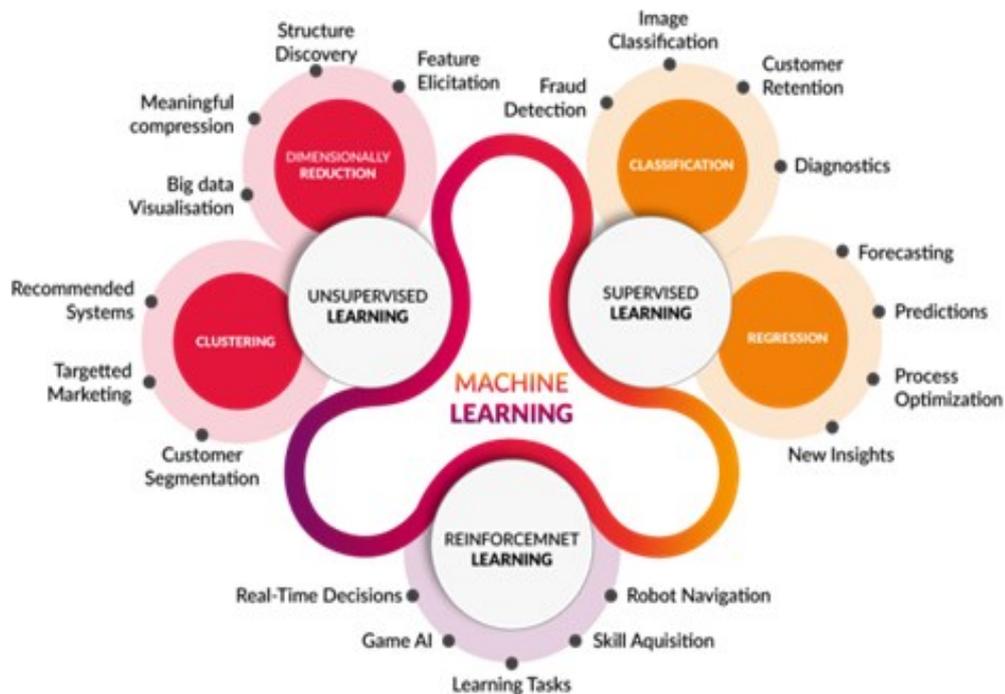


Figura 3 - - Áreas de aprendizagem de ML (PATEL, 2018).

O Aprendizado de Máquina tem uma extensa gama de modelos a serem estudados. Neste capítulo serão apresentados alguns modelos e conceitos porém, este trabalho irá afunilar para uma finalidade específica, pertinente ao trabalho.

2.7.1 Aprendizagem supervisionada

Trata-se de um aprendizado supervisionado quando o modelo, ou algoritmo, é construído a partir de *datasets* (dados de entrada) que são apresentados na forma de pares ordenados (entrada - saída desejada). Esses dados são ditos como dados rotulados, pois é sabido, previamente, a saída esperada para cada entrada de dados (ESCOVEDO; KOSHIYAMA, 2020). Neste caso, os autores simplificam o conceito de aprendizado supervisionado, onde o treinamento do algoritmo consiste em submetê-lo a um número suficiente de exemplos de entradas e saídas desejadas, assim o algoritmo irá criar uma regra geral onde serão mapeadas as entradas nas saídas corretas. A Figura 4 ilustra o modo de trabalho de um algoritmo supervisionado.

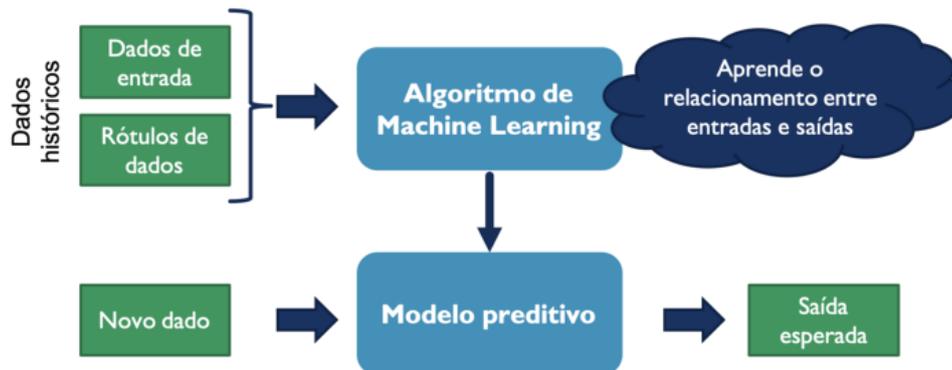


Figura 4 - Aprendizado supervisionado (ESCOVEDO, 2020).

Os problemas para este tipo de aprendizagem podem ser divididos em dois grupos, a Classificação e a Regressão. São exemplos de aprendizagem supervisionadas os seguintes modelos:

- Regressão linear (Linear Regression);
- Regressão logística (Logistic Regression);
- Regressão Adaptativa Múltipla (MARS);
- K-Vizinhos mais próximos (KNN);
- Naïve Bayes;
- Support Vector Machines (SVM);
- Árvore de Decisão (Decision Tree);
- Random Forest;
- Redes Neurais (Neural Networks);

2.7.2 Aprendizagem não supervisionada

Conceituado como o tipo de aprendizagem que se refere à identificação de informações relevantes nos dados sem a presença de um elemento externo para guiar o aprendizado (FACELI et al., 2015). Essencialmente, o aprendizado reside na identificação de propriedades intrínsecas aos dados de entrada, de maneira a construir representações desses dados que possam servir a algum propósito. De uma maneira mais direta, no aprendizado não supervisionado é inexistente a informação dos rótulos, ou seja, não temos a informação de saídas desejadas que serão estimadas pelo algoritmo, sendo assim, chamamos os nossos dados

de não-rotulados (ESCOVEDO; KOSHIYAMA, 2020). Os autores ainda destacam que pelo motivo do algoritmo não receber, durante o treinamento, os resultados esperados, deve descobrir por si só, por meio da exploração dos dados, os possíveis relacionamentos entre eles. Neste caso, o processo de aprendizado busca identificar regularidades entre os dados a fim de agrupá-los ou organizá-los em função das similaridades que apresentam entre si. A Figura 5 ilustra o modo de trabalho de um algoritmo não supervisionado.

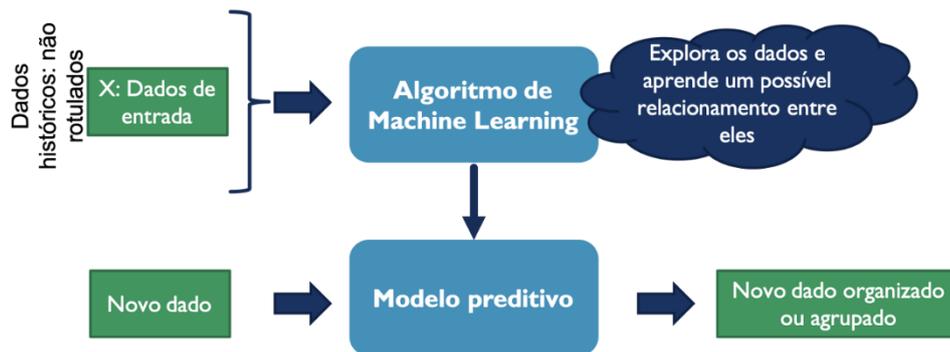


Figura 5 - Aprendizado não supervisionado (ESCOVEDO, 2020).

Os problemas de aprendizagem não supervisionadas, podem ser divididos em dois grupos, a Clusterização (agrupamento) e a Associação. São exemplos de aprendizagem descritivas os seguintes modelos:

- K-means;
- Apriori;
- Análise de Componentes Principais (*Principal Component Analysis* ou PCA);
- Agrupamento hierárquico (*Hierarchical clustering*);
- Análise de Discriminação Linear (*Linear Discriminate Analysis* ou LDA)

2.7.3 Aprendizado de máquina reforçado

Reinforcement learning (RL), ou simplesmente aprendizado por reforço, ensina como tomar as melhores decisões, sequencialmente, dentro de um contexto, para maximizar uma medida de sucesso. Não é dito quais decisões tomar, mas em vez disso deve aprender por si mesmo, experimentando-as. Em linhas gerais, a entidade de tomada de decisão aprende por meio de tentativa e erro (WINDER, 2020), por mecanismos de *feedback* (ou recompensas). A Figura 6 ilustra o modo de trabalho de um algoritmo por reforço.

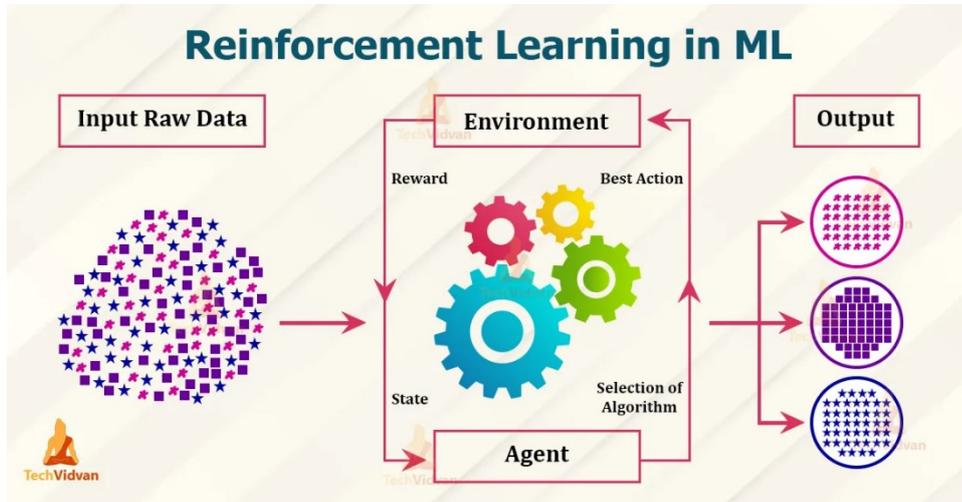


Figura 6 - Aprendizado por reforço (TECHVIDVAN, 2020).

Os problemas de aprendizagem por reforço podem ser baseados em alguns grupos, levando em consideração se são livres de modelo ou baseados em modelo, além de como os algoritmos usam e atualizam suas estratégias, se com políticas ou fora de políticas. São exemplos de aprendizagem descritivas os seguintes modelos:

- Deep Q Network (DQN);
- Deep Deterministic Policy Gradient (DDPG);
- Estado-ação-recompensa-estado-ação (*State-action-reward-state-action* ou SARS);
- Algoritmo Ator-Crítico de Vantagem Assíncrona (*Asynchronous Advantage Actor-Critic Algorithm* ou A3C);

2.7.4 Redes Neurais

Inspirados por uma hipótese de que a atividade mental humana consiste basicamente na atividade eletroquímica em neurônios (células cerebrais), alguns trabalhos mais antigos de Inteligência Artificial tiveram o objetivo de criar redes neurais artificiais (RUSSEL; NORVIG, 2013). Métodos de aprendizagem de Redes Neurais (RN) fornecem uma abordagem robusta para aproximar funções de destino com valor real, valor discreto e valor vetorial. Para certos

tipos de problemas, como aprender a interpretar dados complexos de sensores do mundo real, redes neurais artificiais estão entre os métodos de aprendizagem mais eficazes conhecidos atualmente (MITCHEL, 1997).

Redes Neurais são formadas por nós, ou unidades, de neurônios conectadas por ligações direcionadas. Uma ligação da unidade deve se propagar entre as unidades para a ativação de outra unidade. Cada ligação possui um peso numérico associado a ele, que determina a força e o sinal da conexão (RUSSEL; NORVIG, 2013). Um neurônio matemático, como pode ser descrito um neurônio artificial, similarmente ao natural, recebe um ou mais sinais de entrada e devolve um único sinal de saída, que pode ser propagado em uma rede, como sinal de entrada para um ou mais neurônios da camada posterior (PIO, 2009). As camadas das redes neurais possuem inúmeras arquiteturas, podendo ter uma ou diversas camadas e até mesmo ser divididas em redes que se retroalimentam ou são unidirecionais. Uma rede neural de camada única pode ser descrita como *Perceptron*. As *Multi-layer Perceptron* (MLP), ou redes de múltiplas camadas, são redes que têm uma ou mais camadas ocultas que não são conectadas às saídas da rede. A Figura 7 ilustra alguns exemplos de topologias. Será feita uma breve descrição das arquiteturas mais comuns e mais usadas.

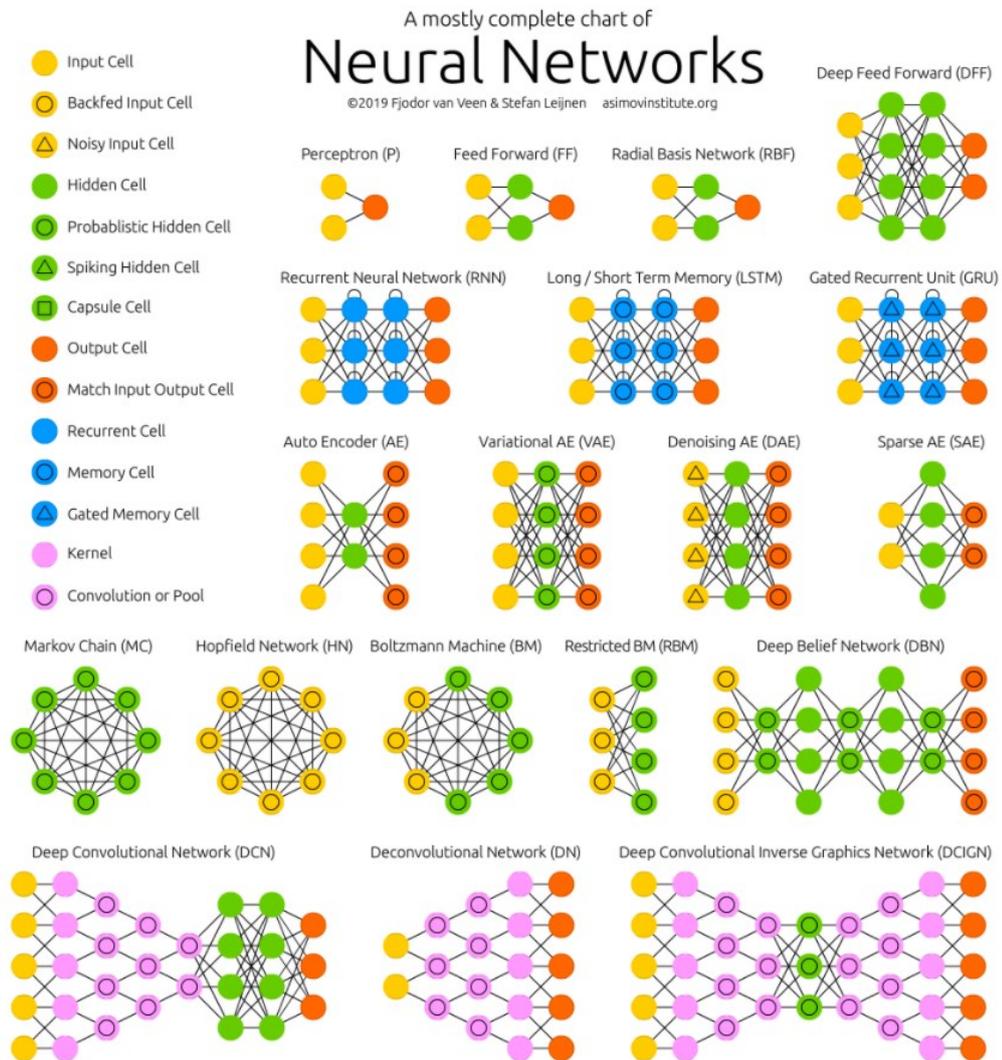


Figura 7 - The neural network zoo (VAN VEEN, 2019)

- **Feed Forward (FF ou FFNN):** Redes neurais simples com alimentação para frente (somente uma direção).
- **Função de base radial (RBF):** São FFNNs com funções de base radial como funções de ativação.
- **Redes Neurais Recorrentes (RNN):** Formam um sistema dinâmico, retroalimentado. A rede alimenta suas saídas de volta às suas próprias entradas.
- **Memória de longo / curto prazo (LSTM):** Redes que tentam combater o problema do gradiente de desaparecimento introduzindo portas e uma célula de

memória definida explicitamente. A porta de entrada determina quanto das informações da camada anterior são armazenadas na célula.

- **Convolutional neural networks (CNN):** As redes neurais convolucionais são bastante diferentes da maioria das outras redes. Elas são usadas principalmente para processamento de imagem, mas também podem ser usadas para outros tipos de entrada, como áudio.

2.7.5 Deep Learning

O *Deep Learning* (DL), ou aprendizagem profunda, é uma tecnologia que utiliza algoritmos mais complexos do que o *Machine Learning* e baseia-se no princípio das RNs, buscando imitar o cérebro humano com ainda mais fidelidade, no que tange à forma de compreender novas informações e gerar resultados a partir delas. (STEFANINI, 2019). Também é conceituada como estruturas que implementam conceitos complexos como uma hierarquia de conceitos mais simples que se relacionam em uma rede profunda de neurônios (GOODFELLOW et al., 2016).

A Figura 8 demonstra a diferença entre o Aprendizado de Máquina clássico e o *Deep Learning*, evidenciando as diferenças. No aprendizado de máquina clássico, é necessário identificar o conjunto de recursos que representam exclusivamente um determinado rosto - por exemplo, a forma arredondada do rosto ou a distância entre os olhos. Em seguida, você aplica um algoritmo classificador de aprendizado de máquina, que aprende a associar um determinado padrão de recursos a uma identidade única. O aprendizado profundo nos permite evitar a engenharia de recursos por completo. Dado "dados rotulados" suficientes (ou seja, imagens de rostos conhecidos) e o ajuste certo, um modelo de aprendizado profundo identificará os recursos mais relevantes dos dados por conta própria (ROBINS, 2020).

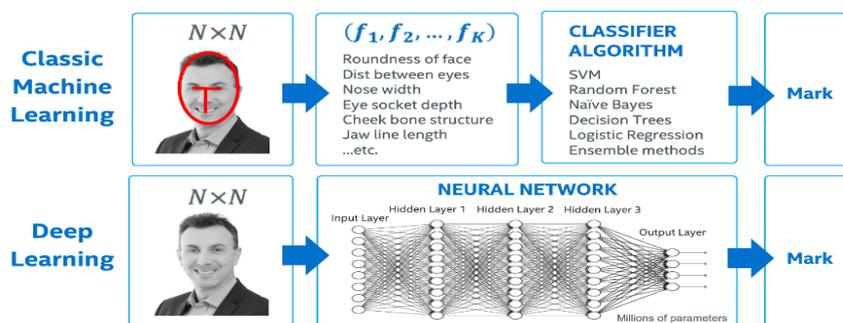


Figura 8 - Aprendizado de máquina vs. aprendizado profundo para reconhecimento facial (ROBINS, 2020)

2.8 SISTEMAS DE RECOMENDAÇÃO

Sistemas de recomendação são filtros de informação para apresentar itens ou objetos – como páginas web, filmes, músicas, livros, medicamentos, lojas, artigos – que provavelmente são do interesse do usuário (SHAFFER, 2001).

Os conceitos básicos de alguns tipos de sistemas de recomendação serão abordados nessa sessão, tendo um pouco mais de destaque para sistemas de recomendação com abordagem clássica de filtragem colaborativa, que será o tipo de sistema utilizado para construção do modelo estudado neste trabalho, já que se trata de uma recomendação baseada na similaridade de histórico escolar.

2.8.1 Sistemas de recomendação abordagem clássica

Um sistema de recomendação (RecSys), em uma abordagem dita como clássica, é uma aplicação de aprendizado de máquina para negócios que, por meio de modelagem de dados e aplicação de algoritmos, tenta prever a nota (*rating*) ou preferência de um usuário a um determinado item. Assim, tais sistemas são classificados de acordo com o tipo de algoritmo usado (MORANDINO; RODRIGUES, 2020). Os principais tipos de recomendação de abordagem clássicas, são:

- **Filtragem Baseada em Conteúdo (*Content-Based*):** Possui como principal vantagem não necessitar de muitos *feedbacks* para recomendar algo “útil”;
- **Filtragem Colaborativa (*Collaborative Filter*):** Gera recomendações que ignoram as características e trabalha por similaridade;
- **Sistemas Híbridos (*Hybrid*):** São os que têm características ou combinam as abordagens de Filtragem Colaborativa e Baseado em Conteúdo.

A Figura 9 visa deixar mais clara a diferença entre os dois tipos de filtragens.

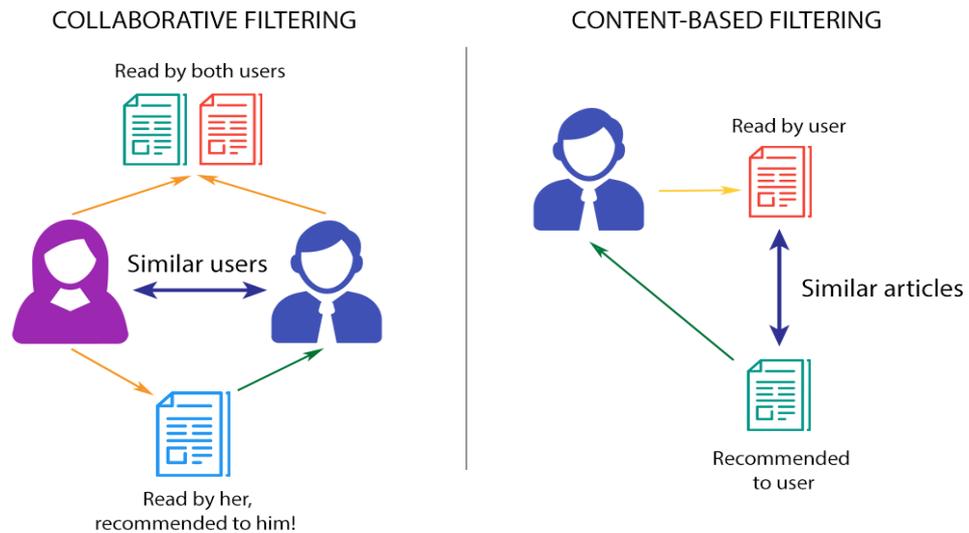


Figura 9 - Filtragem colaborativa e filtragem baseada em conteúdo (MORANDINO; RODRIGUES, 2020)

Os RecSys podem ser utilizados para recomendar conteúdo em diferentes domínios, como livros, música, filme, varejo, notícias etc. Assim como a visão computacional e processamento de linguagem natural foram radicalmente impactadas por DL, atualmente estamos vivenciando o mesmo na área de Sistemas de Recomendação. Não é só a quantidade de *papers* acadêmicos que estão crescendo e avançando o estado-da-arte nos últimos 4 anos. Empresas grandes estão migrando os sistemas clássicos de recomendação para abordagens com DL (SANTANA, 2018).

Ainda existem outros tipos de Sistemas de Recomendação que não se baseiam no modo clássico e sim em vertentes bastante utilizadas, mas não tão difundidas na maioria dos sites e trabalhos que falam sobre o assunto.

2.8.2 Sistemas de recomendação baseado em grafos

Uma representação em grafos tem por objetivo a otimização do processo de recomendação. Pegando o exemplo mais comum, de recomendação de filmes ou obras, temos em um grafo bipartido, onde temos os nodos representando os usuários e itens. Os nodos são interligados por arestas e, desse modo, podemos realizar o agrupamento por alguma medida de similaridade, visando particionar um conjunto de grafos em grupos diferentes que compartilham alguma forma de similaridade (ERRICA et al., 2020).

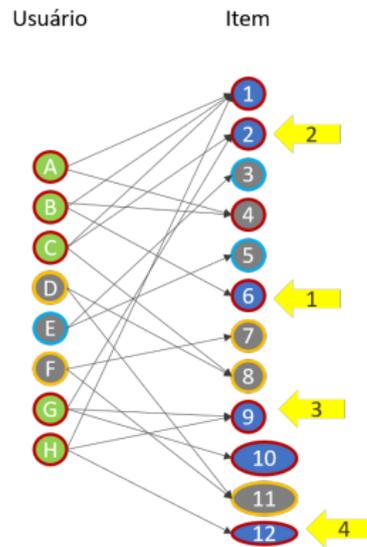


Figura 10 - Representação da ordem de recomendação dos itens para o usuário "A", LEON (2021)

Também é possível criar um modelo com nodos representando usuários, itens e metadados em um grafo, dessa vez, tripartido. A Figura 11 mostra uma ideia desse modelo (SALLES; WILLRICH, 2015).

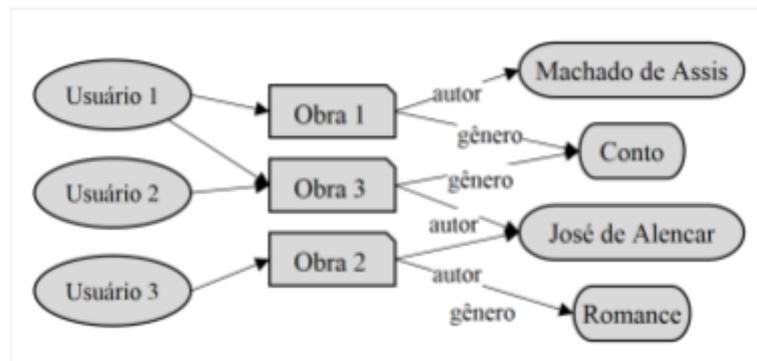


Figura 11 - Recomendação em modelo de grafo, SALLES E WILLRICH (2015)

2.8.3 Sistemas de recomendação baseado em ontologia

As ontologias podem ser definidas, de modo simplificado, como uma representação formalizada e não ambígua do que se conhece de um determinado domínio, definindo-se conceitos, propriedades e relacionamentos (SALLES; WILLRICH, 2015). As ontologias também podem ser vistas como um “catálogo de tipos de coisas” em que se supõe a existência de um determinado domínio, na perspectiva de uma pessoa que usa uma determinada linguagem (SOWA, 1999).

Um sistema de recomendação pode usufruir da modelagem de ontologias que exploram o relacionamento entre conceito e propriedades, podendo obter interesses de usuários que não são diretamente observados (SALLES; WILLRICH, 2015).

Na Figura 12 podemos ter uma noção de como são organizados os dados de conhecimento e informação de uma ontologia.

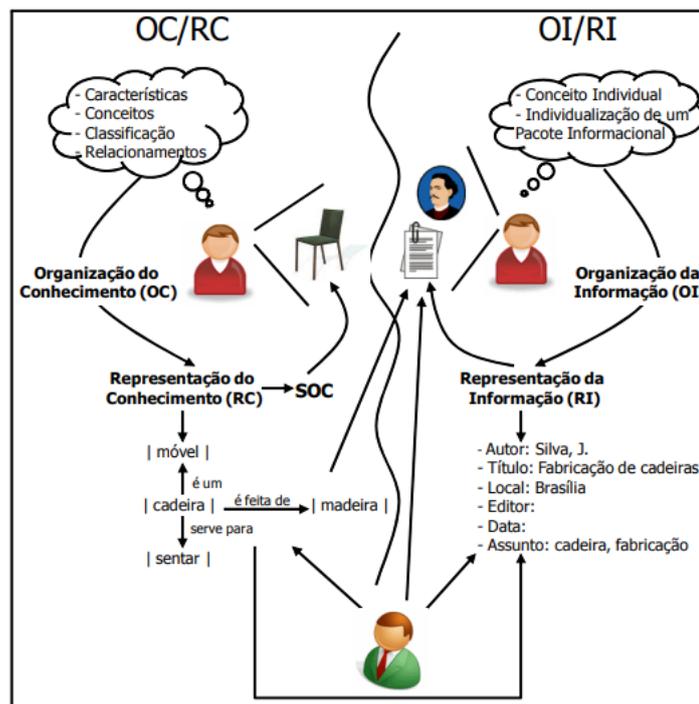


Figura 12 - Organizações de ontologias, Brascher e Café (2008).

Em seu trabalho, Salles e Willrich (2015) exemplificam que para um sistema de recomendação baseado em ontologias, o comportamento de um usuário que acessa várias obras literárias de autores de um determinado estado pode indicar que este usuário teria interesse por obras advindas desse mesmo estado. Então com o auxílio das informações e conhecimentos

contidos em uma ontologia, pode-se realizar inferências das mais diversas formas, podendo usá-las nos sistemas de recomendação.

2.9 BIBLIOTECA FASTAI

Uma *startup* de São Francisco, California, a FastAi, lançou há alguns anos um conjunto de bibliotecas de código projetadas para simplificar radicalmente a escrita de tarefas de aprendizado de máquina. Construído com base na biblioteca PyTorch¹⁴, do *Facebook*, de código aberto, torna mais fácil começar com o aprendizado profundo, permitindo fazer tarefas como executar uma rede neural convolucional para reconhecimento de imagem nos testes de *benchmark* ImageNet com apenas um punhado de linhas de código (RAY, 2018). Seu objetivo é fornecer componentes de alto nível que podem gerar resultados de última geração e fornecer aos pesquisadores componentes de baixo nível que podem ser misturados e combinados para construir novas abordagens. Isso sem comprometer substancialmente a facilidade de uso, flexibilidade ou desempenho. Isso é possível graças a uma arquitetura cuidadosamente dividida em camadas, que expressa padrões subjacentes comuns de muitas técnicas de aprendizado profundo e processamento de dados em termos de abstrações desacopladas. Essas abstrações podem ser expressas de forma concisa e clara, aproveitando o dinamismo da linguagem Python subjacente e a flexibilidade da biblioteca PyTorch (FASTAI, 2019).

Jeremy Howard e Rachel Thomas fundaram a FastAi com o objetivo de democratizar o aprendizado profundo. Eles fazem isso fornecendo um curso online aberto massivo (MOOC) chamado “*Practical Deep Learning for Coders*” (ou Aprendizado prático profundo para codificadores em tradução livre), que não tem outros pré-requisitos, exceto o conhecimento da linguagem de programação Python.

O código da FastAi está disponível em um repositório do GitHub¹⁵ para quem se interessar. Na grande maioria das vezes, o FastAi é usado em plataformas como Google

¹⁴ <https://pytorch.org/>

¹⁵ <https://github.com/fastai/fastai/>

¹⁶<https://colab.research.google.com/notebooks/intro.ipynb> da empresa disponibiliza diversos “*notebooks*” prontos, para facilitar o uso, sempre com textos introdutórios e tutorias. Isso torna a biblioteca prática e fácil de se aprender.

2.10 GOOGLE COLAB E JUPYTER NOTEBOOK

O corriqueiramente chamado apenas de Colab é uma abreviatura para o nome Google Colaboratory. O Google Colaboratory é um ambiente de *notebooks* Jupyter¹⁸ que não requer configuração e é executado na nuvem. Escreva e execute códigos em Python, salve e compartilhe suas análises e acesse poderosos recursos de computação científica, tudo gratuitamente no seu navegador (FIOCRUZ, 2021).

Os chamados *notebooks* do Colab permitem combinar código executável e *rich text* em um só documento, além de imagens, HTML, LaTeX¹⁹ e muito mais. Quando você cria seus próprios *notebooks* do Colab, eles são armazenados na sua conta do Google Drive²⁰. É possível compartilhar os *notebooks* do Colab facilmente com colegas de trabalho ou amigos e permitir que eles façam comentários ou até editem o documento.

Os *notebooks* do Colab são *notebooks* do Jupyter hospedados no Colab. O Jupyter Notebook é um aplicativo da web de código aberto que permite criar e compartilhar documentos que contêm código ativo, equações, visualizações e texto narrativo. Os usos incluem: limpeza e transformação de dados, simulação numérica, modelagem estatística, visualização de dados, aprendizado de máquina e muito mais (JUPYTER, 2021).

Segundo o próprio Google (2021), com o Colab, é possível importar um conjunto de dados de imagem, treinar um classificador de imagens dentro dele e avaliar o modelo, tudo com apenas algumas linhas de código. Os *notebooks* do Colab executam código dos servidores em nuvem do Google. Isso significa que você pode tirar proveito da potência de *hardware* do Google, como GPUs e TPUs, independentemente da potência da sua máquina. Você só precisa de um navegador.

¹⁶ <https://colab.research.google.com/notebooks/intro.ipynb>

¹⁷ <https://docs.fast.ai/>

¹⁸ <https://jupyter.org/>

¹⁹ <https://www.latex-project.org/>

²⁰ <https://www.google.com/intl/pt-br/drive/about.html>

3 TRABALHOS CORRELATOS

Este capítulo tem como objetivo analisar trabalhos correlatos que possam embasar e complementar o estudo feito neste trabalho, no que diz respeito, principalmente, à aprendizagem de máquina aplicada à recomendação de temas de trabalhos de conclusão de curso. Para encontrar os possíveis trabalhos correlatos, foram utilizados alguns passos de Mapeamento Sistemático de Literatura (KITCHENHAM, 2004), adaptado ao contexto de uma pesquisa aplicada de TCC.

3.1 DEFINIÇÃO DO PROTOCOLO DE PESQUISA

A busca pelos trabalhos correlatos tem como objetivo encontrar e analisar os estudos já realizados, trabalhos já produzidos, metodologias e tecnologias utilizadas na literatura atual, que dizem respeito ao aprendizado de máquina no contexto de sistemas de recomendação para temas de trabalhos de conclusão de curso.

Para realização da busca por essa literatura, foi utilizado o mecanismo de buscas do Google Acadêmico²¹, que efetua a busca de produções acadêmicas nos mais diversos repositórios de arquivos acadêmicos. A Tabela 1 mostra os termos, sinônimos e tradução que serão utilizados na busca.

Como critério para o resultado ser considerado relevante, foi considerado que o trabalho deve evidenciar ou uso de aprendizado de máquina, ou de sistema de recomendação, principalmente no contexto dados tabulares ou sistema de recomendação.

Tabela 1 - Termos de busca

Termos	Sinônimos	Tradução (inglês)
sistema de recomendação	sugestões, algoritmos de recomendação	recommendation system, recommender system
dados tabulares	matrizes, dados em tabela, dados tabelados	tabular data

²¹ <https://scholar.google.com.br/>

aprendizado de máquina	aprendizado automático, aprendizagem automática, inteligência artificial	machine learning
------------------------	--	------------------

3.1.1 Critérios de Inclusão e Exclusão:

Para identificar os trabalhos correlatos foram selecionados os estudos que:

- utilizem idioma inglês ou português;
- tenham o conteúdo completo acessível por meio de portal acadêmico;
- apresentem sistemas relacionados com aprendizagem de máquina;
- envolvam a análise de dados tabulares;
- tratem da recomendação ou assuntos similares.

3.2 EXTRAÇÃO DE INFORMAÇÃO

As buscas foram efetuadas em agosto de 2021 utilizando termos individualmente ou em conjunto, com texto normal e também com o termo entre aspas duplas, para restringir os resultados, além de serem realizadas tanto em português quanto em inglês. A Tabela 2 indica os termos buscados e os resultados obtidos. Como o Google Acadêmico retorna os resultados em ordem de relevância, foram analisados os primeiros 100 trabalhos (primeiras 10 páginas) resultantes de cada busca.

Tabela 2 - Resultado da busca

Termos buscados	Resultados	Resultados relevantes	Referências
"recomendação" "trabalhos de conclusão de curso"	3.260	1	(Feitosa et. Al. 2011)
"sistema de recomendação" "dados tabulares"	9	1	(Bonilla, 2020)

recommendation tabular data	83.500	0	
"machine learning" "recommendation"	375.000	0	
"machine learning" "recomendação"	2.760	1	(Trindade, 2020)
machine learning dados tabulares	1.520	0	

3.3 EXTRAÇÃO DE INFORMAÇÃO

Com base nos critérios de inclusão, os estudos selecionados e analisados e que são de alguma forma correlatos a este trabalho, foram considerados relevantes e serão apresentados a seguir.

3.3.1 Recomendação Assistida por Computador para Orientação de Trabalhos de Conclusão de Curso: Baseada no Perfil Informacional e na Capacidade de Orientação Docente

No artigo “Recomendação Assistida por Computador para Orientação de Trabalhos de Conclusão de Curso: Baseada no Perfil Informacional e na Capacidade de Orientação Docente” (FEITOSA et. Al. 2011) os autores problematizam a questão de incompatibilidade de orientando e orientadores, principalmente na modalidade de ensino a distância (EaD) e sugerem um estudo de aprendizado de máquina para um modelo de recomendação, junto com sistema de combinação social, para verificar a compatibilidade entre as duas partes interessadas em um trabalho de conclusão de curso. Foram selecionados alguns alunos e alguns orientadores para fazerem parte do estudo, contando com três diferentes versões de modelos de filtragem para efetuar uma comparação entre eles. São eles: filtragem baseada em desempenho, filtragem baseada em capacidade e modelo informacional de combinação social. Para a validação do sistema de recomendação, os autores escolheram pessoas do curso de bacharelado em Sistemas de Informação, da Universidade Federal de Alagoas, contando com 10 (dez) orientadores em potencial, tendo eles já participado de ao menos 1 (um) módulo de capacitação, além da seleção de 20 (vinte) alunos concluintes de curso, tendo alunos de diferentes Polos na amostra.

Visando a recomendação entre aluno e orientadores, o trabalho contou com a mineração de dados de duas bases de dados convertidas em tabulares, sendo uma delas com os históricos escolares dos alunos e a outra com um questionário respondido por orientadores.

O trabalho contou com a aplicação de algoritmo genético de Holland²² que, em uma síntese simplificada, pode ser descrita como uma ideia de imitar algumas etapas do processo de evolução natural das espécies incorporando-as a um algoritmo computacional.

Autores concluíram que com os estudos feitos e os resultados apresentados, o sistema de recomendação pode ser aplicado para uma amostra muito maior do que a apresentada pelo trabalho. Eles destacam que o fator humano e suas singularidades, são aspectos que se deve ater quando aplicadas técnicas de filtragem baseadas em desempenho e filtragem baseada em capacidade.

Como trabalho futuro eles destacam a questão de aprofundar os estudos nas recomendações baseadas no desempenho do aluno e na capacidade laboral dos professores, bem como a minimização dos problemas que a interação entre professores e professores pode acarretar.

3.3.2 Uma introdução aos sistemas de recomendação, modelos matemáticos, algoritmos e aplicações

A dissertação “Uma introdução aos sistemas de recomendação, modelos matemáticos, algoritmos e aplicações” (BONILLA, 2020) apresenta um resumo, bem elaborado, sobre alguns dos principais modelos de aprendizado de máquina utilizados em sistemas de recomendação ou áreas afins, assim como traz também as métricas utilizadas na avaliação dos métodos. No trabalho, a autora utilizou sistemas de recomendação de filtragem colaborativa e filtragem baseada em conteúdo, tendo utilizado dois tipos de algoritmos para cada sistema de recomendação. As duas duplas são, respectivamente: método dos k-vizinhos mais próximos

²² https://pt.wikipedia.org/wiki/Algoritmo_gen%C3%A9tico

(kNN) junto com decomposição em valores singulares e método dos k-vizinhos mais próximos junto com redes neurais artificiais.

Visando detalhar o uso de cada uma das abordagens de recomendação, o trabalho conta com exemplos pertinentes de uso em empresas tecnológicas, uma síntese histórica por trás do sistema de recomendação e toda a matemática que há em cada um dos algoritmos.

No que diz respeito aos dados que são utilizados no trabalho em questão, são estritamente dados tabulares, podendo conter dados numéricos, categóricos ou em forma de texto. O referido trabalho contou com uma base de dados de uma loja varejista chamada ModCloth, que foi originalmente apresentada por Rishabh Misra, Mengting Wan e Julian McAuley na décima segunda conferência sobre sistemas de recomendações da Association for Computing Machinery (MISRA; WAN; MCAULEY, 2018) sobre sistemas de recomendação. A base de dados conta com avaliações de 47.958 usuários sobre 1.378 itens da loja, tendo a gerado, ao todo, 82.790 avaliações.

Como conclusão e resultado, a autora define, por se tratar de sistemas de recomendação modelados com regressão, uma comparação de valores de erros (erro absoluto médio, erro absoluto médio normalizado, erro quadrático médio e raiz do erro quadrático normalizado) entre os modelos estudados. Ela evidencia os fatores positivos e negativos dos resultados de cada um dos algoritmos, tendo, nesse caso, o sistema de filtro colaborativo parecer apresentar um melhor desempenho, apesar de não se destacar fortemente dos demais algoritmos.

3.3.3 Interpretabilidade em Modelos de Sistemas de Recomendação

A autora da dissertação “Interpretabilidade em Modelos de Sistemas de Recomendação” (TRINDADE, 2020) apresenta em seu trabalho uma proposta de recomendação com o uso de *Deep Learning* interpretável e restrito, precisando desenvolver uma nova arquitetura de rede neural de multicamadas em dois casos de estudo. A questão central do estudo é a interpretabilidade de um modelo de aprendizado de máquina que atua como uma caixa-preta, cujo resultado não pode ser reproduzido de maneira simples, como ocorre com a Aprendizagem Profunda. O trabalho propõe que seja possível utilizar uma

recomendação de caixa-preta com um modo de interpretar seus resultados de forma mais amigável sem comprometer o seu potencial preditivo.

Visando o objetivo de construção de modelo mais interpretável, o trabalho apresenta todo um estudo sobre os sistemas de recomendação, mostrando vantagens e desvantagens, evidenciando os métodos que podem ser utilizados e se debruçando em um sistema de recomendação de ofertas para referenciar o trabalho. O modelo proposto pelo trabalho é, em ambos os casos, com dados tabulares. A autora não deixa explícito qual a sua base de dados e de onde ela foi fornecida, se resumindo apenas a descrever que se trata, numa primeira fase, de um conjunto de dados empresariais e, posteriormente, com dados públicos, com a finalidade do estudo da recomendação personalizada de ofertas e o estudo de previsão de preço de venda em veículos usados.

Para desenvolver o seu trabalho, a autora usa uma estratégia que foca especialmente em arquiteturas rede *Multi-Layer Perceptron* (MLP) ou comumente chamada de Rede Neural Perceptron Multicamadas, tendo a autora realizado uma alteração na arquitetura do modelo.

Como conclusão, a autora traz à tona a questão de modelos de *Deep Learning* serem, cada vez mais, um modelo autônomo e eficaz no processo de otimização, porém sem fornecer uma lógica por trás de suas decisões, faltando transparência do algoritmo para se interpretar o funcionamento interno. O resultado obtido pelo trabalho mostra que a rede construída é capaz de promover ou garantir o comportamento monotônico de variáveis relevantes, sem perder, consideravelmente, o desempenho preditivo. Averiguou-se que a arquitetura não influenciou no ganho de interpretabilidade, podendo haver restrições de monotonicidade.

3.3.4 Considerações Finais

Nos trabalhos selecionados são apresentados diferentes modelos de *Machine Learning* para diferentes contextos e soluções.

Um dos trabalhos apresenta uma visão de compatibilidade entre orientando e orientador, fazendo uso de inteligência artificial para tentar verificar quais professores e alunos possuem mais afinidade para trabalharem em um TCC. Já outro trabalho traz um detalhamento muito rico dos modelos e conceitos de *Machine Learning*, mostrando a parte matemática de alguns modelos e tratando de dados tabulares. E, por último, temos uma dissertação que estuda a interpretabilidade de modelos de recomendação, com uso de *Deep Learning*, ajudando na interpretação de resultados e mostrando quais as vantagens e desvantagens de alguns algoritmos de recomendação.

Este trabalho se diferencia dos supracitados por trazer uma implementação de sistema de recomendação, baseado em modelo tabular de *Machine Learning*, para recomendar temas de TCC aos alunos. Diferente de uns dos trabalhos citados neste capítulo, este sistema de recomendação não terá um formulário com perguntas para onde as respostas se tornarão input para o modelo, bem como não irá recomendar um orientador por afinidade e sim um tema, previamente cadastrado, por algum professor.

No próximo capítulo será abordada a proposta de desenvolvimento de um módulo de recomendação de temas de TCC para o Sistema de Gestão de TCC do INE.

4 DESENVOLVIMENTO

Neste capítulo é apresentada uma análise dos requisitos que foram levantados para implementação do novo módulo, a modelagem dos requisitos na forma de Histórias de Usuário e o desenvolvimento do que será implementado.

4.1 ANÁLISE DE REQUISITOS

Requisitos são declarações que traduzem ou expressam uma necessidade e suas restrições e condições associadas (ISO/IEC/IEEE 12207:2017)²³.

Os requisitos para a nova funcionalidade foram levantados: (i) por meio de entrevista não estruturada com o orientador e o autor deste trabalho, ambos usuários afetados pelo trabalho que será realizado e; (ii) a partir de análise de trabalhos correlatos.

Os requisitos funcionais são representados na Tabela 3.

Tabela 3 - Requisitos funcionais

Requisito Funcional	Descrição
RF01	O sistema deve permitir que o orientador cadastre (CRUD) áreas de pesquisa;
RF02	O sistema deve permitir que o orientador cadastre (CRUD) propostas de temas de projetos de TCC;
RF03	O sistema deve exibir, em uma página do perfil de professor, uma lista com suas propostas de temas cadastrados;
RF04	O sistema deve exibir, na página inicial uma lista com os temas cadastrados pelos orientadores;
RF05	O sistema deve exibir os TCCs atualmente em orientação e os recentemente finalizados pelo orientador;

²³ Systems and software engineering--Software life cycle processes

RF06	O sistema deve permitir a recomendação de tema de TCC para um aluno;
RF07	O sistema deve mostrar em tela uma lista de temas/áreas de pesquisas e os professores interessados nesses temas;
RF08	O sistema deve apresentar informações sobre os dados que serão usados para fazer a recomendação;
RF09	O sistema deve permitir vincular uma proposta de tema, proposto pelo professor, no momento de cadastro de um TCC;
RF10	O sistema deve exigir o cadastro de área e de tipo de projeto no momento de cadastro de um TCC.

4.1.1 HISTÓRIAS DE USUÁRIO

Os requisitos desse trabalho são analisados por meio de histórias de usuário. Uma história de usuário pode ser definida como (1) uma narrativa simples ilustrando um requisito do usuário da perspectiva de uma persona (ISO/IEC/IEEE 26515: 2018)²⁴ (2) uma descrição narrativa de um requisito, função, recurso ou atributo de qualidade de *software*, apresentada como uma narrativa das interações desejadas do usuário com um sistema de *software*²⁵.

US01. Como orientador eu preciso definir temas de projetos de TCC para que os alunos interessados possam conhecer as minhas áreas de pesquisa.

Requisitos funcionais: RF01, RF02, RF03.

Protótipo de tela:

²⁴ Systems and software engineering: Developing information for users in an agile environment, 3.16

²⁵ Software Extension to the PMBOK(R) Guide Fifth Edition

Tela de cadastro de card com proposta de projeto para TCC

Cadastrar proposta de projeto

Área do projeto Campo de texto ou combobox

Tema do projeto Campo de texto ou combobox

Breve descrição do projeto Campo de texto com limite de caracteres

Link para mais detalhes Campo de confirmação de senha

Cancelar Cadastrar

Figura 13 - Protótipo US01

A Figura 13 apresenta um protótipo de tela para cadastro de temas de TCC das áreas de pesquisa do professor, onde serão apresentados os campos para inserir uma área de pesquisa, um tema para projeto de TCC, uma descrição simples do que se trata a proposta e ainda um campo para inserção de link onde o aluno pode ter mais detalhes sobre a proposta. O campo de área de pesquisa será apresentado em campo de seleção com as áreas já cadastradas, tentando evitar duplicidade, mas permitindo a inserção de nova área.

Critérios de aceitação

- US01.C1 - O tema do projeto é cadastrado com sucesso;
- US01.C2 - O tema do projeto aparece na listagem para os alunos;
- US01.C3 - É possível alterar ou excluir os temas de projetos cadastrados pelo orientador.

US02. Como aluno eu preciso visualizar as propostas de temas de projetos de TCC cadastradas pelos professores para que eu possa verificar se um tema e/ou um professor se enquadram em um projeto que me interessa.

Requisitos funcionais: RF04, RF05.

Protótipo de tela:

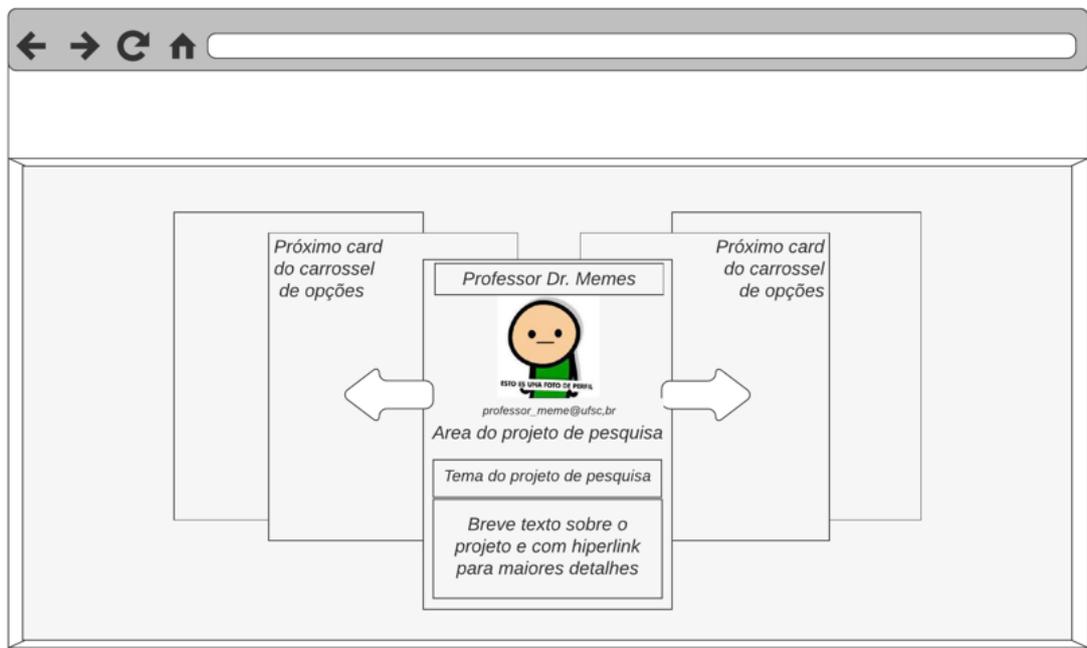


Figura 14 - Protótipo US02

Uma lista de propostas de temas de projetos de TCC é exibida na tela inicial do usuário aluno, como mostra o protótipo de tela da Figura 14. Nela são apresentados os detalhes da proposta, como área de pesquisa, tema, breve descrição e o professor. Ao clicar no *hiperlink* o aluno será redirecionado para uma nova aba/página do navegador para a página previamente cadastrada pelo professor.

Critérios de aceitação

- US02.C1 - O tema do projeto cadastrado pelo professor aparece na tela;

- US02.C2 - As informações de área de pesquisa, tema e descrição aparecem na tela;
- US02.C3 - É possível ver mais de uma proposta, quando houver mais de um cadastro pelos professores.

US03. Como aluno eu preciso poder fazer uma busca de recomendação para temas de TCC, para que eu possa ver quais os três temas que mais se enquadram com o meu perfil acadêmico.

Requisitos funcionais: RF06, RF07, RF08.

Protótipo de tela:

Aviso sobre as informações consumidas para realizar a recomendação e qualquer outro tipo de aviso pertinente para acesso aos dados e recomendação.

Concordo em usar os dados do CAGR da minha matrícula para realizar a recomendação de tema de TCC.

Iniciar recomendação

<p><i>Professor Dr. Memes</i> e-mail: professor_meme@ufsc.br Link Lattes: http://lattes.cnpq.br/123</p> <p style="text-align: center;"> </p>	<p><i>Informações sobre os temas retornados na recomendação</i></p>
<p><i>Professor Dr. Memes</i> e-mail: professor_meme@ufsc.br Link Lattes: http://lattes.cnpq.br/123</p> <p style="text-align: center;"> </p>	<p><i>Informações sobre os temas retornados na recomendação</i></p>
<p><i>Professor Dr. Memes</i> e-mail: professor_meme@ufsc.br Link Lattes: http://lattes.cnpq.br/123</p> <p style="text-align: center;"> </p>	<p><i>Informações sobre os temas retornados na recomendação</i></p>

Figura 15 - Protótipo US03

O protótipo de tela da recomendação de temas para TCC, Figura 15, apresenta um aviso de que serão usados os dados do histórico escolar no CAGR do aluno para realizar a recomendação; um campo de *checkbox* para o consentimento do aluno para uso dos dados; botão para iniciar o algoritmo de recomendação e campo com lista dos três primeiros ranqueados pela recomendação.

Critérios de aceitação

- US03.C1 - A área da lista de temas recomendados deve ser preenchida com, no mínimo um e no máximo 3; temas de projetos;
- US03.C2 - As informações de tema e do professor, deve aparecer na área de lista de temas recomendados;

US04. Como aluno eu preciso poder, no momento do cadastro do meu TCC, cadastrar uma proposta de projeto, uma área do projeto e o tipo de projeto.

Requisitos funcionais: RF09, RF10.

Protótipo de tela:

O protótipo de tela apresenta um formulário com os seguintes elementos:

- Título:** Um campo de texto grande e vazio.
- Descrição:** Um campo de texto grande e vazio.
- Responsável:** Um menu suspenso com o texto "Selecione o responsável".
- Proposta de projeto:** Um menu suspenso.
- Área do projeto:** Um menu suspenso.
- Tipo de projeto:** Um menu suspenso.
- Botão Salvar:** Um botão localizado no canto inferior direito do formulário.

Figura 16 - Protótipo US04

O protótipo de tela do cadastro de TCC 16, apresenta os campos já existentes, de professor responsável, título e resumo do projeto, onde também serão apresentadas as opções, obrigatórias, de área do projeto e tipo de projeto. Já a opção de tema proposto pelo orientador será opcional.

Critérios de aceitação

- US04.C1 - Os campos de área do projeto e tipo do projeto precisam ser preenchidos para poder cadastrar o TCC;
- US04.C2 – Uma lista de áreas deve ser exibida para que uma delas seja selecionada;
- US04.C3 – Uma lista de tipos de projetos deve ser exibida para que uma delas seja selecionada;
- US04.C4 – Uma lista de propostas de projetos deve ser exibida para que uma delas possa ser selecionada;

4.2 IMPLEMENTAÇÃO

Neste capítulo será abordado o processo de desenvolvimento e implementação relacionados ao novo módulo e às novas telas, propostas neste trabalho, para o Sistema de Gestão de TCC do INE. Também será apresentada uma seção com as tecnologias utilizadas pelo sistema, bem como outra seção com os passos e particularidades inerentes ao processo de configuração de ambiente de desenvolvimento do sistema.

4.2.1 Seleção de tecnologias

Nessa seção são apresentadas as tecnologias escolhidas, e sua motivação, para realizar a implementação do novo módulo proposto. Primeiramente será descrita a sequência de passos realizados para configuração do ambiente de desenvolvimento do sistema.

A maioria das tecnologias é restrita pelos *frameworks* utilizados pela SETIC²⁶ (Superintendência de Governança Eletrônica e Tecnologia da Informação e Comunicação) que foram utilizados na versão atual do sistema de gerenciamento de TCCs do INE.

No que diz respeito à tecnologia utilizada no aprendizado de máquina, empregado para realizar a recomendação de temas, foi escolhida a biblioteca para linguagem Python²⁷, FastAI²⁸, por sua facilidade de implementação e por fornecer uma interface consistente para as aplicações mais comuns de *Machine Learning*, além de um prévio conhecimento existente.

Como tecnologias e ferramentas utilizadas no ambiente de desenvolvimento tem-se o Spring, um *framework* com o intuito de tornar as aplicações mais simples, ágeis e flexíveis; o JavaServer Faces, um *framework* para aplicações WEB que irá auxiliar nas interfaces com usuário; o Hibernate que auxilia e facilita a manipulação dos dados na base de dados; o Maven²⁹, como gerenciador de dependências; o MySQL como opção para base de dados e o MySQL WorkBench³⁰, como ferramenta auxiliar na gerência da base de dados.

4.2.2 Preparação do ambiente

Para que fosse possível realizar a implementação do novo módulo para o Sistema de Gestão de TCC do INE, foi necessário, primeiramente, preparar todo ambiente de desenvolvimento.

Com intuito de simplificar e facilitar o desenvolvimento, recomenda-se utilizar um IDE (ambiente integrado de desenvolvimento) que vem sendo utilizado pelos alunos e professores que trabalharam no Sistema de Gestão de TCC do INE, o IntelliJ³¹. O IDE IntelliJ fornece um suporte a grande parte das tecnologias envolvidas e ferramentas utilizadas para o desenvolvimento necessário.

Uma visão macro do processo de configuração pode ser observada na Figura 17, que posteriormente será detalhado.

²⁶ <https://setic.ufsc.br/>

²⁷ <https://www.python.org/about/>

²⁸ <https://docs.fast.ai/>

²⁹ <http://maven.apache.org/>

³⁰ <https://www.mysql.com/products/workbench/>

³¹ <https://www.jetbrains.com/pt-br/idea/>

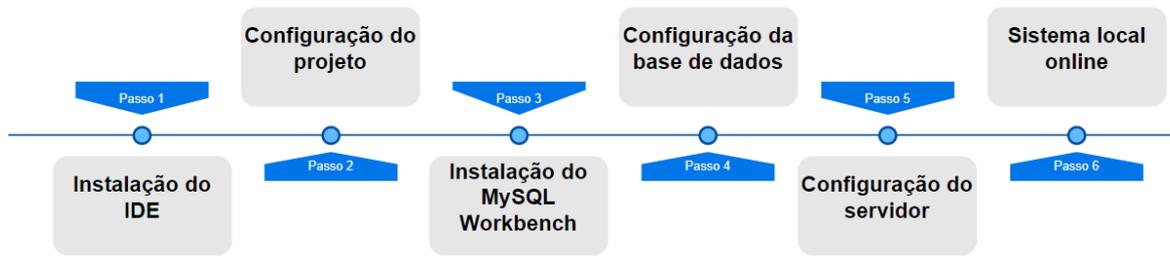


Figura 17 - Fluxograma de configuração de ambiente de desenvolvimento.

Como primeiro passo, fez-se a instalação do IDE de modo padrão e, após a conclusão da instalação, iniciou-se a importação e configuração do projeto.

O código do projeto do Sistema de Gestão de TCC do INE está hospedado em um repositório no GitLab³², sistema gerenciador de repositórios de código fonte, o que facilitou em todo o processo de configuração, já que é possível, no IDE IntelliJ, iniciar um novo projeto a partir da importação de código fonte de um repositório do GitLab. Sendo assim, foi acessado o repositório Gitlab onde encontra-se a última versão do código fonte do Sistemas de TCC do INE e coletada a URL para importação do projeto no IntelliJ.

Como o Sistema de Gestão de TCC do INE utiliza o Maven para o gerenciamento das dependências do projeto, na pasta raiz encontra-se um arquivo chamado pom.xml³³ onde são definidas as dependências do projeto, que é reconhecido pelo IDE pois o IntelliJ possui um suporte nativo ao Maven. A própria ferramenta, levando em conta o conteúdo desse arquivo, irá se encarregar de carregar as dependências a partir de uma central de repositórios e adicioná-la ao projeto.

Sendo assim, após a realização da importação do projeto e o reconhecimento do arquivo pom.xml pelo IDE, as dependências foram carregadas automaticamente. A Figura 18 mostra a estrutura criada após a importação do projeto junto com uma pequena parte do arquivo pom.xml.

³² <https://about.gitlab.com/>

³³ <https://maven.apache.org/guides/introduction/introduction-to-the-pom.html>

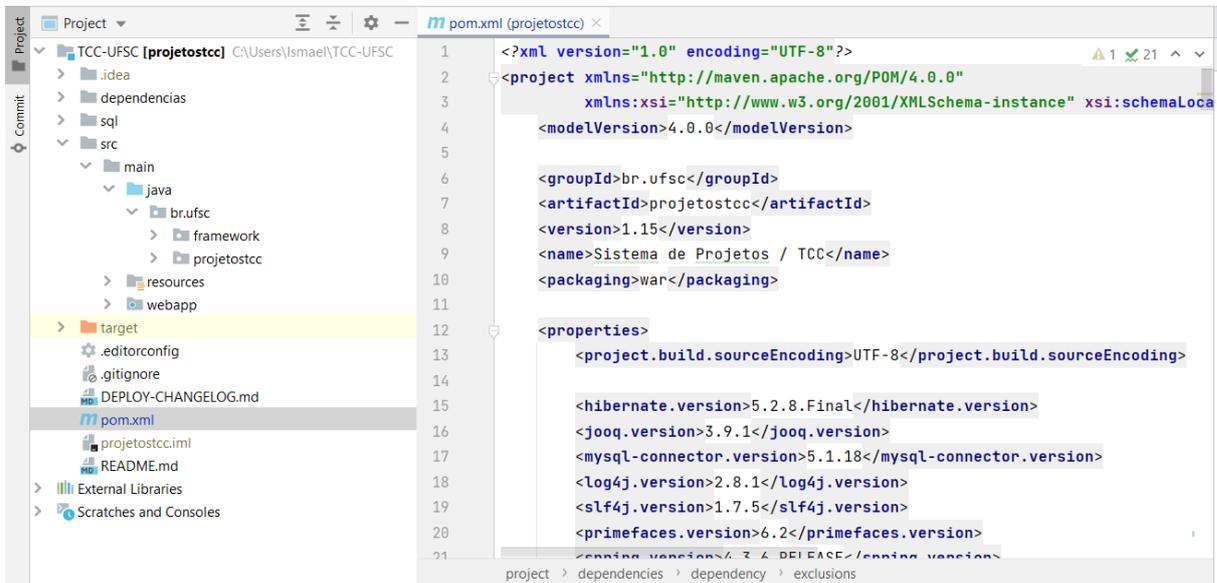


Figura 18 - Organização do projeto e arquivo pom.xml.

Após a configuração do projeto no IDE, foi preciso realizar a instalação e configuração das ferramentas relacionadas à base de dados. Primeiramente instalou-se o MySQL versão 5.5.56, que é a versão utilizada atualmente no projeto em produção. Logo após, foi instalado o MySQL WorkBench, que facilitou a configuração da base de dados por possuir uma interface mais amigável, a partir de um arquivo de *backup* disponível no mesmo repositório, do GitLab, onde se encontra o código fonte do sistema. O MySQL WorkBench além de auxiliar na configuração da base de dados, também é útil na definição de estrutura e mostrando os dados da base, possibilitando a visualização das tabelas e seus conteúdos. A Figura 19 ilustra a ferramenta com parte da estrutura da base de dados e uma tabela de exemplo.

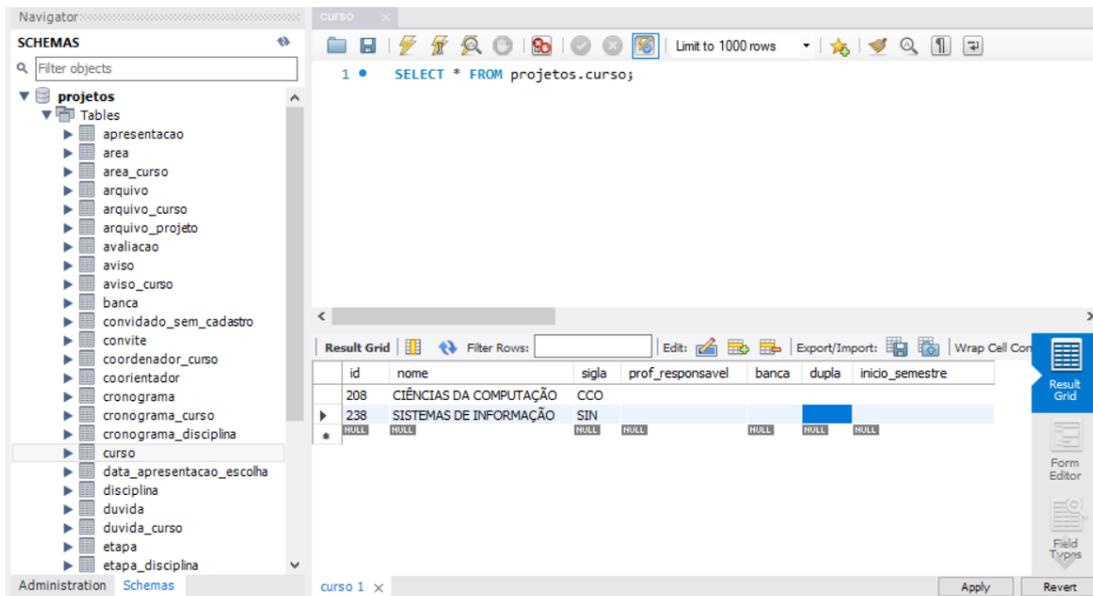


Figura 19 - Organização da base de dados e exemplo de tabela de dados.

Em uma próxima etapa, foi necessário fazer alguns ajustes em arquivos do código fonte, configurações no próprio sistema operacional do computador e também no servidor de aplicação usado no IDE.

O Sistema de Gestão de TCCs do INE utiliza, como servidor de aplicação, o Apache Tomcat³⁴ que, apesar de ser um recurso nativo do IntelliJ, precisa ser baixado configurado para pleno funcionamento da aplicação no ambiente local. Para tal, precisa referenciar o diretório raiz do projeto na hora da criação da configuração de execução no IDE, bem como alterar detalhes no arquivo server.xml e inserir alguns arquivos no diretório de configuração do Tomcat. Com isso serão possíveis conexões HTTP utilizando SSL e o Sistemas de TCCs do INE, em sua versão de desenvolvimento, vai autenticar no sistema de autenticação centralizada (CAS) da UFSC. Assim, as seguintes configurações foram realizadas:

- baixar o Apache Tomcat versão 8³⁵;
- colocar o arquivo server.keystore, presente na pasta raiz do repositório do GitLab do projeto, na pasta conf³⁶ do Tomcat baixado e descompactado;
- nessa mesma pasta, alterar o arquivo server.xml adicionando a tag abaixo:

³⁴ <http://tomcat.apache.org/>

³⁵ <https://tomcat.apache.org/download-80.cgi>

³⁶ "C:\Users\<usuario>\apache-tomcat-8.5.65\conf"

```

    <Connector SSLEnabled="true" clientAuth="false"
keystoreFile="conf/server.keystore" keystorePass="padrao" maxThreads="150"
port="8443" protocol="HTTP/1.1" scheme="https" secure="true"
sslProtocol="TLS"/>

```

- fazer o redirecionamento de host no sistema operacional, neste caso, com o uso do Windows, fez-se com a edição do arquivo³⁷ em modo de administrador e inserindo uma linha com o texto “127.0.0.1 projetostcc” sem aspas;
- se a senha *root* configurada na sua base de dados for diferente de "root", precisa alterar no arquivo "application-dev.properties" (no IntelliJ clicar “ctrl+n”, clicar em "files" na janela aberta e buscar pelo arquivo) a linha `database.password=root`

E, por último, é necessário realizar uma configuração do Apache Tomcat dentro do IDE IntelliJ, que nos fornece um *template* para alterarmos, tornando a configuração mais simples. Nesse *template* é preciso ser definida a pasta raiz do Tomcat e os artefatos que serão implantados. Ainda que exista a possibilidade de alteração de outros detalhes, isso não se faz necessário pois o está como *default* no *template* já é suficiente. A Figura 20 mostra os detalhes dessa configuração do Tomcat no IntelliJ.

³⁷ “C:\Windows\System32\drivers\etc\hosts”

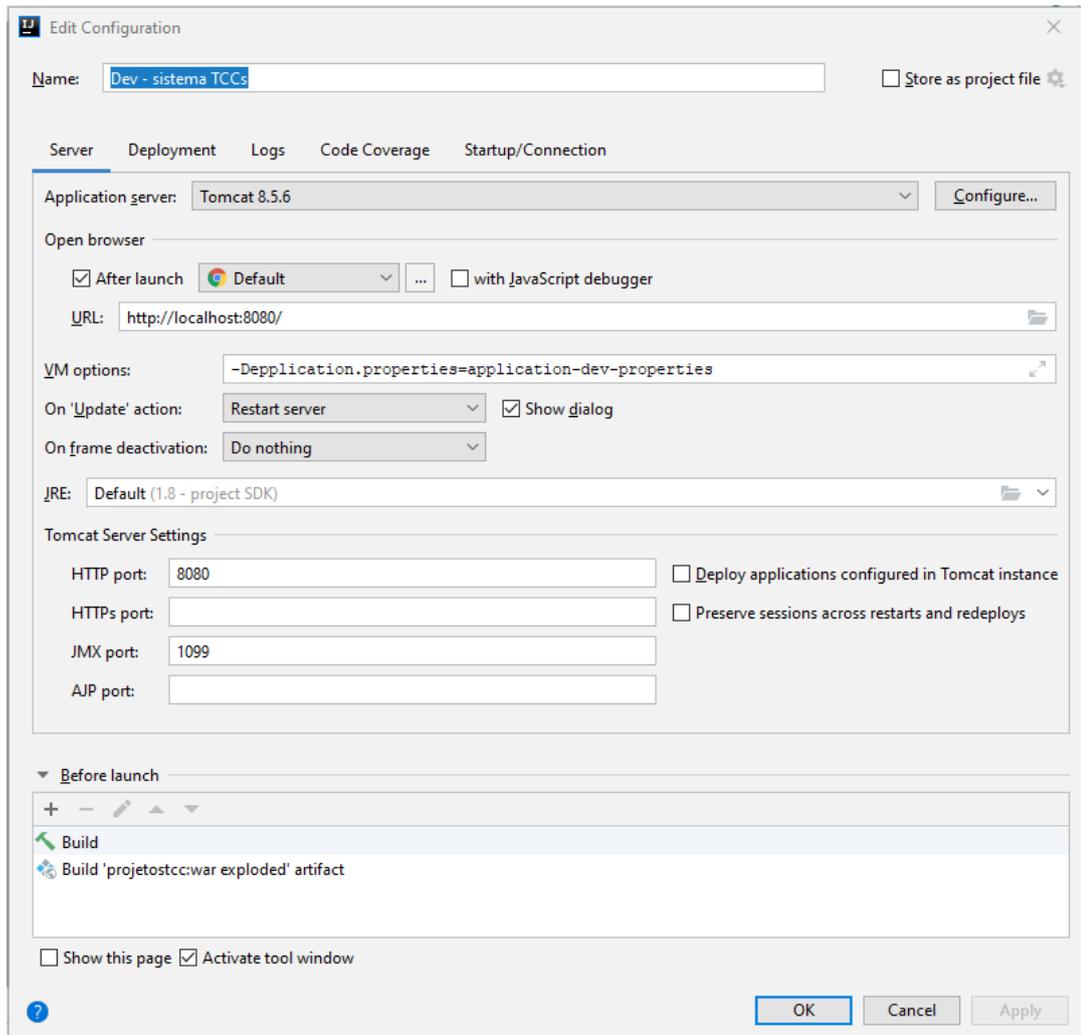


Figura 20 - Configuração utilizada em ambiente de desenvolvimento

Após realizar todos esses passos, o ambiente de desenvolvimento local está pronto para rodar. Isso pode ser feito com um simples clique no botão “Run” do IDE.

4.2.3 Preparação do conjunto de dados

Antes da preparação do conjunto de dados é necessário saber qual o resultado esperado do modelo de recomendação, bem como qual a abordagem adotada, pois tais informações nortearão toda a escolha e preparação dos dados. Como tem-se o objetivo de recomendar uma área de estudo, ou seja, realizar a classificação de uma variável categórica usando os dados de

histórico escolar de alunos já formados, será usado uma abordagem clássica de filtragem colaborativa. A tabela de dados terá uma tupla de registro para cada usuário, que é definido por conter um número de matrícula, que seria seu identificador, dados relacionados ao seu TCC e notas de histórico escolar. Tendo essas informações de forma clara, pode-se começar a extrair e preparar os dados.

Para que o modelo de recomendação fosse treinado com os dados de notas do histórico escolar dos alunos, foi necessário a extração de tais dados de um segundo sistema, o CAGR. Na base de dados do Sistema de Gestão de TCCs do INE tem-se os dados de “id_ufsc”, número de identificação único para cada aluno vinculado a UFSC e matrícula dos alunos cadastrados no sistema. Porém nem todos os alunos cadastrados no sistema possuem o dado de matrícula. Dessa forma, criou-se uma query para realizar a extração de uma lista de “ids” para alunos que já concluíram a graduação e possuem nota na disciplina de Projetos II, para o curso de Sistemas de Informação, ou Trabalho de Conclusão de Curso II (TCC) para o curso de Ciências da Computação. De posse desta lista de “ids” o código do Sistema de Gestão de TCCs do INE foi alterado, temporariamente, para buscar, através da biblioteca já existente no sistema, as matrículas vinculadas a cada “id”, necessitando filtrar por matrícula de graduação e dos dois cursos estudados no trabalho. Desse modo, criou-se uma nova lista com os dados de matrículas dos alunos onde, para cada matrícula, foi realizada a busca do histórico escolar no CAGR. Em um primeiro momento os dados de histórico de cada matrícula foram extraídos de forma bruta, com a tabela contendo todas as notas de todas as disciplinas cursadas por aquele aluno. Como existiam muitos alunos com notas em diferentes disciplinas, de diferentes áreas, sem que houvesse uma homogeneidade das disciplinas cursadas, dado que alguns alunos eram mais antigos e demoraram mais do que o comum para concluírem o TCC, optou-se por transformar os dados. No lugar das notas brutas, foi calculada uma média de notas para cada área de pesquisa, que contém um conjunto de disciplinas, de modo que o conjunto de dados, que continha mais de duzentas colunas de notas, uma coluna para cada disciplina, foi reduzido para apenas traze, uma coluna para cada área. As áreas em questão foram extraídas e adaptadas da listagem de GPDA³⁸, Grupo de Professores de Disciplinas Afins, do INE.

Tabela 4 - Tabela de áreas para classificação de TCCs

AREA

³⁸ <https://planos.inf.ufsc.br/modulos/gpdas/>

Ciência, Tecnologia, Sociedade e Empreendedorismo
Computação Numérica
Engenharia de Software, Gerência de Projetos
Fundamentos de Programação
Fundamentos Matemáticos
Introdução à Computação (ICC)
Organização de Computadores e Sistemas Embarcados
Probabilidade e Estatística
Projetos, Estágios e Intercâmbios
Redes de Computadores e Segurança
Sistemas Operacionais, Concorrência e Paralelismo
Teoria da Computação e Linguagens
Tratamento e Banco de Dados

Tabela 5 - Tabela com disciplinas da área de Tratamento e Banco de Dados

Área 13 - Tratamento e Banco de Dados
INE5225 - Fundamentos de Sistemas de Banco de Dados
INE5423 - Banco de Dados I
INE5431 - Sistemas Multimídia
INE5432 - Banco de Dados II
INE5454 - Tópicos Especiais em Gerência de Dados
INE5600 - Bancos de Dados III
INE5613 - Bancos de Dados I
INE5616 - Bancos de Dados II
INE5623 - Projeto de Banco de Dados
INE5639 - Sistemas Multimídia
INE5643 - Data Warehouse
INE5644 - Data Mining
INE5647 - Qualidade da Informação
INE5650 - Web Semântica, Ontologias e Sistemas de Informação
INE5662 - Tópicos Avançados em Sistemas de Informação II

Ainda houve a necessidade da criação de classificação, que foi realizada de forma manual, pelo autor do presente trabalho, para cada um dos TCCs contidos no conjunto de dados, pois tal classificação, até então inexistente, seria usada para treinamento do modelo de recomendação. Cada TCC foi classificado quanto a área do projeto, uma disciplina que mais se

aproxima com o trabalho e, posteriormente, do tipo de trabalho. Na Figura 21 temos um recorte do *dataset* para ilustrar como ficou estruturado:

E	F	G	H	I	J	K	L	M
AREA	ORIENTADOR	CLASSIF_AREA	DIA_NC	Area1	Area2	Area3	Area4	Area5
Tratamento e Banco de Dados	Carina Friedrich Dorneles	INE5644 - Data Mining	9.00	7.50	8.34	7.50	8.00	7.50
Tratamento e Banco de Dados	Renato Fileto	INE5644 - Data Mining	9.00	8.50	8.42	8.00	9.00	8.00
Tratamento e Banco de Dados	Ronaldo dos Santos Mello	INE5644 - Data Mining	9.00	9.75	6.50	7.90	8.13	8.34
Tratamento e Banco de Dados	Ronaldo dos Santos Mello	INE5647 - Qualidade da In	9.00	8.75	6.00	7.25	7.63	6.84
Tratamento e Banco de Dados	Elder Rizzon Santos	INE5650 - Web Semântica	9.00	8.84	7.60	6.90	7.50	8.50
Tratamento e Banco de Dados	Carina Friedrich Dorneles	INE5650 - Web Semântica	9.00	9.25	7.50	7.90	8.00	6.50

Figura 21 - Recorte do *dataset* com dados estruturados

Outro ponto importante que merece ser destacado é que o modelo precisa ser treinado com dados faltantes. Como no *dataset* temos as *features* “orientador” e “classf_area”, que é uma disciplina contida na área do TCC, ou seja, uma especialização ou subárea do TCC, para auxiliar na precisão da recomendação, alguns desses dados foram excluídos manualmente. Isso se deve porque no momento da solicitação de recomendação o aluno pode querer gerar a recomendação baseada apenas no histórico, sem prever qualquer possível orientador ou disciplina que mais o agradou no curso. Então o modelo precisa conseguir recomendar mesmo sem esses dados.

4.2.4 Treinamento do modelo

Para os testes iniciais de treinamento de modelo, primeiramente foi estudada a maneira mais adequada para realizar a extração de dados, já que é necessário obter dados das bases do Sistema de Gestão de TCC do INE e do CAGR da UFSC. Após realizar tal extração, os dados foram tratados para possibilitar o treinamento e os testes de diferentes modelos.

Em uma das primeiras tentativas foi avaliado um modelo de filtragem colaborativa, onde cada linha da tabela indicava um aluno, ou como é comumente usado, um usuário, e usando como *feature* apenas as médias das notas de cada área de disciplinas pertinentes dos cursos do INE.

	Area1	Area2	Area3	Area4	Area5	Area6	Area7	Area8	Area9	Area10
0	8.50	7.00	7.34	9.17	8.13	8.00	7.75	7.50	8.50	6.84
1	7.75	6.50	8.00	7.17	7.67	8.50	7.34	8.00	9.63	7.50
2	9.00	7.40	6.60	8.50	8.00	7.50	7.25	8.63	8.10	7.67
3	8.67	7.25	7.30	7.00	6.00	6.00	7.50	8.17	7.50	7.50
4	8.50	6.00	6.80	8.13	7.00	7.00	8.50	6.00	8.40	6.50
..
321	8.50	6.00	6.40	6.67	6.34	7.50	6.75	6.00	9.10	6.17
322	8.50	6.50	6.88	6.80	7.17	8.00	6.75	6.75	9.20	7.63
323	6.00	6.67	7.50	6.75	6.00	6.75	9.00	8.00	7.00	6.34
324	7.20	7.84	7.00	7.50	6.50	6.50	7.67	6.34	7.50	7.17
325	9.75	6.50	7.63	6.50	6.63	8.00	7.25	7.00	9.38	7.50

Figura 22 - Recorte do *dataset* importado no código Python

Com o auxílio de uma biblioteca foi realizado o cálculo de uma matriz de distância entre pares, usando como parâmetro o cosseno. Logo após é chamada uma função de recomendação onde é passado como parâmetro um *dataset*, um valor “n”, que será a quantidade de dados de resposta da função, e o número da linha, que seria o equivalente ao “id” do aluno, para verificar os valores recomendados para esse aluno. No exemplo da figura 23 temos como retorno com o “id” e o valor de distância para os cinco dados mais próximos do dado passado como parâmetro.

```
[ ] # Índices dos conteúdos originais
indices = pandas.Series(range(len(dataset_filtrado)), index=dataset_filtrado.index)
indices

# Função que obtém recomendações com base na pontuação de similaridade de cosseno
def get_recommender(idx, dataset_filtrado, top_n = 5):
    sim_idx = indices[idx]
    sim_scores = list(enumerate(cosine_sim[sim_idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:top_n+1]
    idx_rec = [i[0] for i in sim_scores]
    idx_sim = [i[1] for i in sim_scores]

    return indices.iloc[idx_rec].index, idx_sim

get_recommender(321, dataset_filtrado, top_n = 5)

(Int64Index([279, 57, 281, 121, 149], dtype='int64'),
 [0.9981383650294038,
 0.9980173158673651,
 0.9978902093848087,
 0.9977206582364168,
 0.997664755959038])
```

Figura 23 – Imagem com recorte de trecho do código com função de recomendação

Porém, neste caso, consegue-se apenas verificar os dados mais próximos, sem conseguir recomendar uma área ou tema de projeto.

Em um segundo momento, foi testado outro modelo que utilizava um algoritmo, SVD e KNN, para tentar realizar a recomendação. Porém o modelo também não satisfazia a necessidade.

```

algo = SVD(random_state = 42)
algo.fit(train)
pred = algo.test(test)

# RMSE = Raiz quadrada do erro-médio
# MAE = erro absoluto médio

accuracy.rmse(pred)

uid = str(16105151.0)
iid = str(13)

# predição para usuário específico
pred = algo.predict(uid, iid, r_ui=4, verbose=True)

RMSE: 1.7643
user: 16105151.0 item: 13      r_ui = 4.00  est = 5.47  {'was_impossible': False}

```

Figura 24 - Trecho de código com modelo usando SVD

Depois de avaliar alguns modelos chegou-se em um mais adequado, um modelo de classificação tabular, usando a biblioteca FastAi para facilitar a implementação.

Modelo de classificação tabular:

Para treinar o modelo de recomendação é criado um ambiente de desenvolvimento em nuvem, com o auxílio da ferramenta Google Colab. No ambiente do Colab são inseridos código Python, em um Jupyter *notebook*, que definirão a lógica de programação responsável pela importação dos dados, classificação das *features*, treinamento do modelo e testes. Como está sendo utilizada uma biblioteca que facilita a utilização e implementação de ML e, conseqüentemente, de sistema de recomendação, os exemplos disponibilizados pela FastAi nortearam a lógica implementada. Como a estrutura dos dados utilizados é de planilha, ou seja, dados em uma tabela, será utilizado como base um modelo de classificação de dados tabulares, adaptando e alterando o código para a necessidade da estrutura e objetivo do presente trabalho.

Para a implementação do modelo são utilizadas algumas bibliotecas de Python, sejam elas para conectar ao local que hospeda a planilha com os dados, para plotar gráficos no ambiente Colab ou até mesmo para importar o modelo de recomendação.

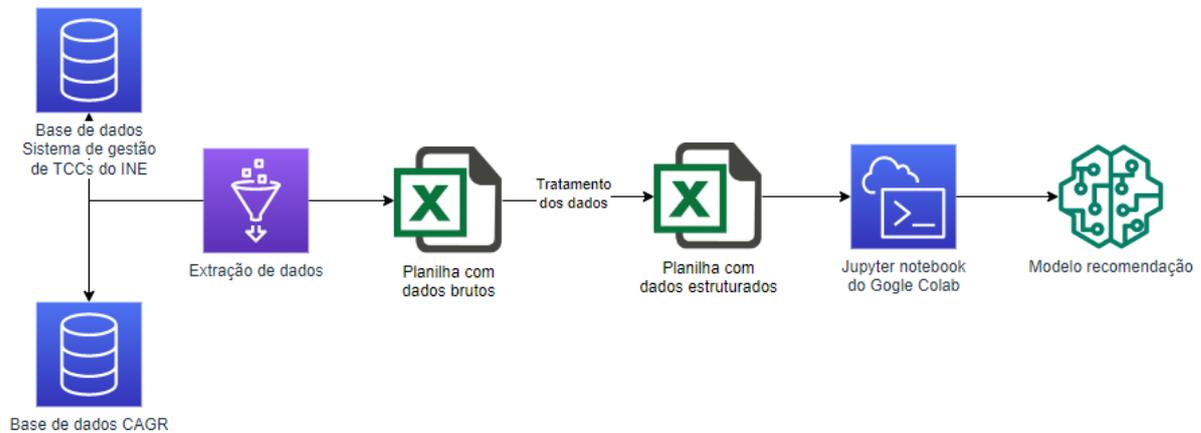


Figura 25 - Arquitetura de desenvolvimento do modelo de recomendação

Em um primeiro passo a planilha hospedada do Google Drive foi importada para o *notebook* do Colab para que os dados pudessem ser utilizados no modelo. Nem todas as *features*, ou colunas, foram utilizadas no modelo, sendo necessária a exclusão de determinadas colunas. Outra necessidade é a de classificação das *features*, de acordo com o tipo de variável, separando em variáveis contínuas, basicamente dados numéricos, e variáveis categóricas. A Figura 26 mostra a separação realizada:

```
[ ] cat_names = ['CURSO', 'ORIENTADOR', 'CLASSIF_AREA']
    cont_names = ['Area1', 'Area2', 'Area3', 'Area4', 'Area5', 'Area6',
                  'Area7', 'Area8', 'Area9', 'Area10', 'Area11', 'Area12', 'Area13']
    procs = [Categorify, FillMissing, Normalize]
    y_names = 'AREA'
    y_block = CategoryBlock()
```

Figura 26 - Trecho do código com separação de variáveis e variável alvo

A Figura 26 também mostra a nossa variável alvo em “*y_names*”, *feature* que possui a área de um TCC na planilha e que será o dado recomendado pelo modelo. Também é necessário realizar a divisão do *dataset* para que parte dele seja usada no treinamento do modelo e parte para o teste. A biblioteca do FastAi já realiza a separação de forma simples, deixando 80% dos dados para treinamento e os outros 20% para teste. Essa separação é

realizada de forma aleatória. Na Figura 27 pode-se observar o valor 309 para o “len(df)” que seria o tamanho do *dataset*, além da divisão do *dataset* com os dados aleatórios, onde o valor “#61” corresponde aos 20% de teste e o valor “#248” corresponde aos 80% de treinamento.

```

[16] splits = RandomSplitter()(range_of(df))
      splits

      ((#248) [58,182,235,205,84,278,226,99,239,71...],
       (#61) [81,195,291,72,46,196,79,17,15,214...])

[17] range_of(df)[:5], len(df)

      ([0, 1, 2, 3, 4], 309)
  
```

Figura 27 - Trecho do código com divisão de *dataset*.

Para realização da classificação tabular, com FastAi, é usada função que implementa, implicitamente, uma rede neural como modelo de IA. Para o treinamento em questão, foi usado como métrica a acurácia. A Figura 28 mostra como é criado, propriamente dito, o modelo de classificação e o gráfico onde é vista a perda em relação as taxas de aprendizado do modelo.

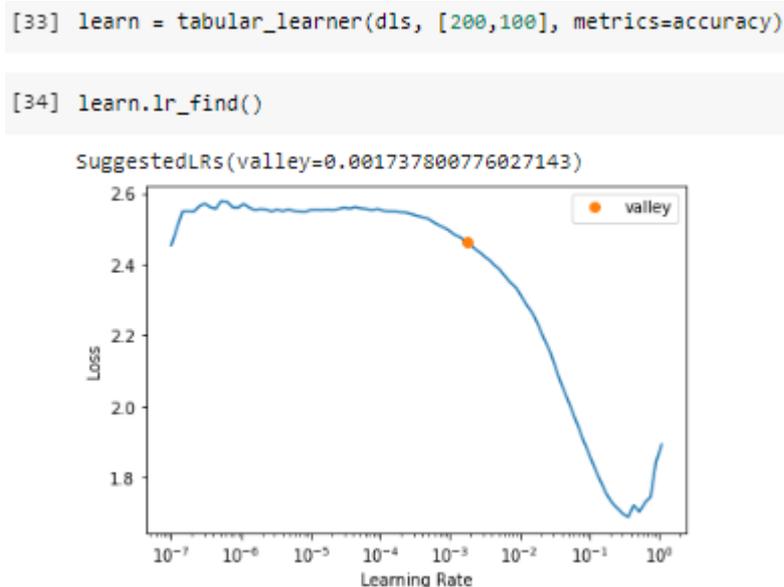


Figura 28 - Trecho do código com criação do modelo e gráfico de perda x aprendizado

E então o modelo é treinado, chamando a função “fit” e passando como parâmetro o número de épocas que se quer iterar e o valor de taxa de aprendizagem, dada em escala

logarítmica. O valor de taxa de aprendizagem é sugerido no gráfico da Figura 28, onde, neste exemplo, seria o valor 10^{-3} . No treinamento de um modelo de ML, pode-se escolher as métricas de avaliação que serão observadas. As métricas de avaliação derivam das classificações realizadas de uma tabela chamada Matriz de Confusão, onde são comparadas as respostas do modelo com as classificações corretas (verdadeiro positivo, verdadeiro negativo, falso positivo e falso negativo), e são calculadas a partir dos erros e acertos do modelo comparado aos resultados esperados. Como exemplos de métricas temos a precisão, que irá validar quantos resultados preditos como positivos são realmente positivos, a especificidade validando a porcentagem de amostras negativas corretamente identificadas. Temos também a acurácia, que diz respeito a porcentagem de elementos corretamente classificados, sejam eles positivos ou negativos, e indica uma performance mais geral do modelo. A acurácia é a métrica que será utilizada no modelo desenvolvido neste trabalho.

A Figura 29 mostra, como exemplo, teste realizado com dez épocas e o valor recomendado no gráfico acima. Também pode-se observar os valores de outras duas métricas, perda de treinamento (*train_loss*) e perda de validação (*valid_loss*) que são, respectivamente, a avaliação de como o modelo se ajusta aos dados de treinamento e avaliação do modelo no conjunto de dados reservados para validação.

```
[42] learn.fit(10, 1e-3)
```

epoch	train_loss	valid_loss	accuracy	time
0	0.007995	0.725165	0.819672	00:00
1	0.010325	0.714480	0.819672	00:00
2	0.009294	0.717439	0.819672	00:00
3	0.008914	0.727555	0.819672	00:00
4	0.008176	0.717966	0.819672	00:00
5	0.008080	0.717007	0.819672	00:00
6	0.007824	0.716961	0.819672	00:00
7	0.007543	0.716562	0.819672	00:00
8	0.007261	0.709829	0.819672	00:00
9	0.007478	0.706748	0.819672	00:00

Figura 29 - Trecho de código com recorde das épocas do treinamento do modelo

Vale lembrar que por se tratar de uma divisão aleatória do *dataset*, entre dados para treinamento e dados para validação, sempre que o código é executado obtém-se resultados diferentes para o aprendizado do modelo, mesmo que na maioria das vezes essa diferença seja desprezível. Porém, se nessa divisão do *dataset* uma das partes ficar com muitos índices, linhas do *dataset*, com dados faltantes para as *features* de “orientador” e “classif_area”, os valores de taxa de aprendizado e o número necessários de épocas para um resultado melhor pode variar. Após o treinamento realizado, há a necessidade de validar a classificação realizada pelo modelo treinado. Isso é feito chamando a função de predição em lote da biblioteca. Na Figura 30 é ilustrado como é chamada a predição em lote e ainda são plotadas na tela as primeiras predições realizadas com o conjunto de dados de teste:

```
[37] # predição em lote
d1 = learn.d1s.test_d1(df.iloc[:100])

d1.show_batch()
```

	CURSO	ORIENTADOR	CLASSIF_AREA	Area1	Area2	Area3	Area4	Area5	Area6	Area7	Area8	Area9	Area10	Area11	Area12	Area13	AREA
0	SISTEMAS DE INFORMAÇÃO (noturno)	ORIENTADOR DEFAULT	DISCIPLINA DEFAULT	1.00	1.000000e+00	1.00	1.00	1.00	1.00	1.00	1.00	1.000000e+00	1.00	1.00	1.000000e+00	1.000000e+00	AREA DEFAULT
1	CIÊNCIAS DA COMPUTAÇÃO	Joao Candido Lima Dovicchi	INE5655 - Gestão Estratégica de Tecnologia, Informação e Comunicação	8.84	9.845508e-09	7.67	7.67	6.25	8.50	6.00	6.75	-1.555664e-07	7.00	7.00	7.500000e+00	8.000000e+00	Ciência, Tecnologia, Sociedade e Empreendedorismo
2	SISTEMAS DE INFORMAÇÃO (noturno)	José Eduardo de Lucca	INE5428 - Informática e Sociedade	7.30	6.750000e+00	6.00	6.50	6.50	8.25	8.00	7.50	7.670000e+00	6.50	6.63	-1.565468e-07	-4.693868e-08	Ciência, Tecnologia, Sociedade e Empreendedorismo
3	SISTEMAS DE INFORMAÇÃO (noturno)	José Eduardo de Lucca	INE5428 - Informática e Sociedade	8.00	7.340000e+00	8.00	7.50	7.50	7.25	9.75	8.38	8.170000e+00	6.50	7.59	-1.565468e-07	-4.693868e-08	Ciência, Tecnologia, Sociedade e Empreendedorismo

Figura 30 - Trecho do código com recorde de predição realizada

E, por último, realizamos a validação dos dados recomendados e posteriormente a exportação, em arquivo, do modelo treinado. Com isso, o modelo pode ser importado, usando o mesmo pacote da biblioteca FastAi, em qualquer outro sistema. A Figura 31 mostra o processo de validação do modelo e como ele é exportado:

```
[39] learn.validate(d1=d1)

(#2) [0.0609070286154747,0.9800000190734863]

[40] learn.export('modeloParaPredicao.pk1')
```

Figura 31 - Trecho do código com valores de validação e exportação do modelo.

4.2.5 Implementação de modelo de recomendação

Após o modelo estar pronto é necessário exportá-lo para ser usado junto ao Sistema de Gestão de TCCs do INE. Como a utilização de código Python dentro de um sistema desenvolvido todo em Java não seria algo simples, optou-se por criar um micro serviço que faria a parte da recomendação, com o código em Python, tendo a chamada via API como mostra a Figura 32:

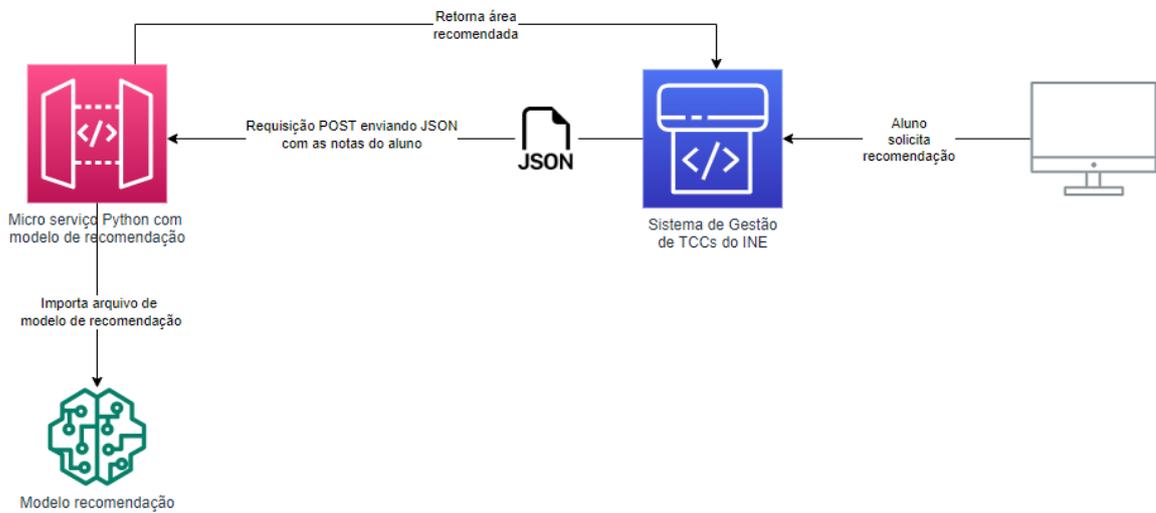


Figura 32 - Arquitetura do serviço de recomendação.

Com o modelo treinado e extraído, tendo um micro serviço respondendo de maneira satisfatória, pode-se realizar a implementação de código para executar a recomendação e mostrar o resultado em tela. Junto com o desenvolvimento do módulo de recomendação, foram realizados testes da implementação de código, tanto de tela quanto de regra de negócios, dos *cards* de propostas de projetos e da associação entre um TCC, no momento que é cadastrado, com um tema de projeto proposto pelo professor.

O código fonte do Sistema de Gestão de TCCs do INE, até então, não prevê testes unitários e estes não foram implementados. Porém todo código criado foi testado, de forma manual, de acordo com regra de negócios e de boas práticas, de forma exaustiva, a fim de capturar qualquer eventual erro, seja em tela ou em comportamento.

Durante os testes foi necessário realizar a alteração de estrutura da atual base de dados, seja das tabelas já existentes ou das tabelas criadas, ao longo do desenvolvimento, bem como incrementar a lógica em diversos pontos do código já existente. Também houve a necessidade de testar a comunicação entre o código fonte e o micro serviço de recomendação através de uma API, tendo que validar os dados de entrada e os dados de saída dessa chamada.

Com o modelo treinado e extraído, tendo um micro serviço respondendo de maneira satisfatória, pode-se realizar a implementação de código para executar a recomendação e mostrar o resultado em tela. Junto com o desenvolvimento do módulo de recomendação, foram realizados testes da implementação de código, tanto de tela quanto de regra de negócios, dos *cards* de propostas de projetos e da associação entre um TCC, no momento em que é cadastrado, com um tema de projeto proposto pelo professor.

O código fonte do Sistema de Gestão de TCCs do INE, até então, não prevê testes unitários e estes não foram implementados. Porém todo código criado foi testado, de forma manual, de acordo com regra de negócios e de boas práticas, de forma exaustiva, a fim de capturar qualquer eventual erro, seja em tela ou em comportamento.

Durante os testes foi necessário realizar a alteração de estrutura da atual base de dados, seja das tabelas já existentes ou das tabelas criadas, ao longo do desenvolvimento, bem como incremento de lógica em diversos pontos já existentes do código. Também houve a necessidade de testar a comunicação entre o código fonte e o micro serviço de recomendação através de uma API, tendo que validar os dados de entrada e os dados de saída dessa chamada.

4.2.6 Avaliação do modelo de recomendação

Nesta etapa, o objetivo é avaliar a correção das recomendações de área de projeto para TCCs. No código de criação do modelo de recomendação o *dataset* é dividido entre dados para treinamento e dados para teste. Com isso pode-se observar uma primeira validação, no próprio código do modelo, como ilustra a Figura 21.

```
[41] # predição em lote
      dl = learn.dls.test_dl(df.iloc[:100])
```

Figura 33 - Trecho do código que realiza predição do *dataset* de teste.

No treinamento do modelo foram realizados testes com diversos números de épocas, verificando que com 20 épocas se tem um resultado satisfatório, mas tendo um pouco de ganho na acurácia e no valor de perda com 30 épocas. Porém, quando se aumenta esse número de épocas, é observado uma oscilação e um efeito contrário, onde o valor de perda aumenta, a acurácia diminui e posteriormente chega em um resultado muito próximo do alcançado com 30 épocas, não fazendo sentido deixar o modelo com mais do que isso. Assim, com esse número de épocas, foi possível obter o valor bem próximo a 0,85 para a acurácia, o que indica um bom resultado, já que a acurácia está bem próxima de 1, que seria o valor ideal. Também se notou um valor próximo de 0,1 para o *train_loss*, que representa a diferença entre o valor de saída da rede e o esperado para uma determinada entrada. A diminuição gradativa do valor de *loss* a cada *epoch* indica que a rede está obtendo mais sucessos ao fazer previsões. Neste caso, quando mais próximo de 0, melhor.

26	0.109608	0.770053	0.828125	00:00
27	0.100589	0.780865	0.828125	00:00
28	0.092078	0.812015	0.828125	00:00
29	0.085185	0.800105	0.843750	00:00

Figura 34 – Ilustração mostrando na coluna 1 o número de épocas, na coluna 2 a perda e na coluna 4 a eficácia.

Logo após foram feitos diversos testes manuais, com a tabela com dados conhecidos que, posteriormente, foram parcialmente apagados e inseridos como *input* para o sistema de recomendação. A Figura 35 mostra a tabela com os dados completos:

CURSO	COD_AREA	AREA	ORIENTADOR	CLASSIF_AREA
CIÊNCIAS DA COMPUTAÇÃO		Redes de Computadores e Segurança	Carlos Becker Westphall	INE5429 - Segurança em Computação
CIÊNCIAS DA COMPUTAÇÃO		Redes de Computadores e Segurança	Carlos Becker Westphall	INE5429 - Segurança em Computação
SISTEMAS DE INFORMAÇÃO (noturno)		Engenharia de Software, Gerência de Projetos	Fabiane Barreto Vavassori Benitti	INE5614 - Engenharia de Software
CIÊNCIAS DA COMPUTAÇÃO		Teoria da Computação e Linguagens	Christiane Anneliese Gresse Von Wange	INE5624 - Engenharia de Usabilidade
CIÊNCIAS DA COMPUTAÇÃO		Engenharia de Software, Gerência de Projetos	Christiane Anneliese Gresse Von Wange	INE5624 - Engenharia de Usabilidade
CIÊNCIAS DA COMPUTAÇÃO		Fundamentos de Programação	Frank Augusto Siqueira	INE5646 - Programação para Web

Figura 35 - Planilha com dados conhecidos.

Já a Figura 36 mostra a mesma tabela, porém depois que alguns dados foram excluídos e a Figura 37 mostra a mesma tabela depois de ser importada no código de teste da recomendação.

C	D	E	F	G
CURSO	COD_AREA	AREA	ORIENTADOR	CLASSIF_AREA
CIÊNCIAS DA COMPUTAÇÃO		AREA DEFAULT		INE5429 - Segurança em Computação
CIÊNCIAS DA COMPUTAÇÃO		AREA DEFAULT	Carlos Becker Westphall	
SISTEMAS DE INFORMAÇÃO (noturno)		AREA DEFAULT		
CIÊNCIAS DA COMPUTAÇÃO		AREA DEFAULT		INE5624 - Engenharia de Usabilidade
CIÊNCIAS DA COMPUTAÇÃO		AREA DEFAULT	Frank Augusto Siqueira	

Figura 36 – Recorte de planilha com dados conhecidos e posteriormente excluídos.

```
[3] path = pathlib.Path('/content/drive/My Drive/TCC - Sistema recomendação INE')
df_test = pd.read_excel(path/'tabela_modelo_medias_v4_test.xlsx')
df_test = df_test.drop(columns=['ID_USFC', 'MATRICULA', 'COD_AREA'])
df_test.head(n=8)
```

	CURSO	AREA	ORIENTADOR	CLASSIF_AREA	MEDIA_NOTA	Area1	Area2	Area3	Area4	Area5	Area6	Area7	Area8	Area9	Area10	Area11	Area12	Area13
0	CIÊNCIAS DA COMPUTAÇÃO	AREA DEFAULT	NaN	INE5429 - Segurança em Computação	10.0	7.50	6.0	7.5	7.75	7.67	10.0	7.25	6.75	8.20	7.17	6.88	7.34	8.00
1	CIÊNCIAS DA COMPUTAÇÃO	AREA DEFAULT	Carlos Becker Westphall	NaN	10.0	9.50	6.5	7.0	7.50	8.13	6.5	7.84	8.50	10.00	7.67	7.88	8.10	7.84
2	SISTEMAS DE INFORMAÇÃO (noturno)	AREA DEFAULT	NaN	NaN	8.5	9.00	7.4	6.6	8.50	8.00	7.5	7.25	8.63	8.10	7.67	6.50	7.50	0.00
3	CIÊNCIAS DA COMPUTAÇÃO	AREA DEFAULT	NaN	NaN	9.5	8.50	6.0	8.2	7.67	8.00	8.0	7.00	7.50	8.84	7.50	8.13	8.22	8.00
4	CIÊNCIAS DA COMPUTAÇÃO	AREA DEFAULT	NaN	INE5624 - Engenharia de Usabilidade	10.0	8.75	6.5	7.1	6.84	7.50	8.5	6.75	7.50	9.13	7.00	6.75	7.34	8.13
5	CIÊNCIAS DA COMPUTAÇÃO	AREA DEFAULT	Frank Augusto Siqueira	NaN	7.5	8.50	6.0	7.0	6.25	6.50	8.0	7.25	6.00	8.10	7.50	6.50	6.40	7.00

Figura 37 - Planilha com dados excluídos importada no código do modelo.

Para cada caso conhecido da planilha foi gerada uma recomendação. Apesar de, nesse caso, a recomendação ter acertado todos os resultados, nem em todos os testes isso ocorreu e não é garantido que o modelo sempre acertará um dado conhecido. A Figura 38 ilustra o último teste realizado com uma tabela que possuía 20 tuplas de dados conhecidos. Nela pode-se observar que dos 20 dados, apenas 2, os destacados em vermelho, o modelo não conseguiu prever o valor correto.

AREA	AREA RECOMENDADA	ORIENTADOR	CLASSIF_AREA
Redes de Computadores e Segurança	Redes de Computadores e Segurança	Carlos Becker Westphall	INE5429 - Segurança em Computação
Redes de Computadores e Segurança	Redes de Computadores e Segurança	Carlos Becker Westphall	INE5429 - Segurança em Computação
Engenharia de Software, Gerência de Projetos	Engenharia de Software, Gerência de Projetos	Fabiane Barreto Vavassori Benitti	INE5614 - Engenharia de Software
Teoria da Computação e Linguagens	Teoria da Computação e Linguagens		INE5624 - Engenharia de Usabilidade
Engenharia de Software, Gerência de Projetos	Engenharia de Software, Gerência de Projetos	Christiane Anneliese Gresse Von Wangenheim	
Fundamentos de Programação	Fundamentos de Programação	Frank Augusto Siqueira	INE5646 - Programação para Web
Tratamento e Banco de Dados	Engenharia de Software, Gerência de Projetos		
Tratamento e Banco de Dados	Tratamento e Banco de Dados		INE5650 - Web Semântica, Ontologias e Sistemas de Informação
Fundamentos de Programação	Fundamentos de Programação		
Teoria da Computação e Linguagens	Engenharia de Software, Gerência de Projetos		
Tratamento e Banco de Dados	Tratamento e Banco de Dados		
Ciência, Tecnologia, Sociedade e Empreendedorismo	Ciência, Tecnologia, Sociedade e Empreendedorismo	Elder Rizzon Santos	INE5428 - Informática e Sociedade
Tratamento e Banco de Dados	Tratamento e Banco de Dados	Ronaldo dos Santos Mello	INE5454 - Tópicos Especiais em Gerência de Dados
Tratamento e Banco de Dados	Tratamento e Banco de Dados		
Engenharia de Software, Gerência de Projetos	Engenharia de Software, Gerência de Projetos	Patricia Vilain	INE5455 - Testes de Software
Engenharia de Software, Gerência de Projetos	Engenharia de Software, Gerência de Projetos	Patricia Vilain	INE5455 - Testes de Software
Teoria da Computação e Linguagens	Teoria da Computação e Linguagens	Elder Rizzon Santos	
Teoria da Computação e Linguagens	Teoria da Computação e Linguagens		INE5633 - Sistemas Inteligentes
Engenharia de Software, Gerência de Projetos	Engenharia de Software, Gerência de Projetos	Jean Carlo Rossa Hauck	INE5614 - Engenharia de Software

Figura 38 - Recorte de tabela usada para validação da recomendação

Isso se dá pelo peso do treinamento para cada *feature* (coluna da tabela com determinado dado) para cada tupla de dados. Para determinado aluno as notas podem ter peso maior do que um possível orientador, já em outro caso, o peso da coluna de orientador pode se sobressair ao peso do histórico escolar. A Figura 39 mostra os resultados da recomendação realizada no código, com a área recomendada na direita da imagem:

```

CURSO ORIENTADOR CLASSIF_AREA Area1 Area2 Area3 Area4 Area5 Area6 Area7 Area8 Area9 Area10 Area11 Area12 Area13 AREA
0 CIÊNCIAS DA COMPUTAÇÃO #na# #na# 8.5 6.0 8.2 7.67 8.0 8.0 7.0 7.5 8.84 7.5 8.13 8.22 8.0 Teoria da Computação e Linguagens
tensor(9)

row, cls, probs = learn_segundo.predict(df_test.iloc[4])
row.show()
cls
CURSO ORIENTADOR CLASSIF_AREA Area1 Area2 Area3 Area4 Area5 Area6 Area7 Area8 Area9 Area10 Area11 Area12 Area13 AREA
0 CIÊNCIAS DA COMPUTAÇÃO #na# INE5624 - Engenharia de Usabilidade 8.75 6.5 7.1 6.84 7.5 8.5 6.75 7.5 9.13 7.0 6.75 7.34 8.13 Engenharia de Software. Gerência de Projetos
tensor(2)

row, cls, probs = learn_segundo.predict(df_test.iloc[5])
row.show()
cls
CURSO ORIENTADOR CLASSIF_AREA Area1 Area2 Area3 Area4 Area5 Area6 Area7 Area8 Area9 Area10 Area11 Area12 Area13 AREA
0 CIÊNCIAS DA COMPUTAÇÃO Frank Augusto Siqueira #na# 8.5 6.0 7.0 6.25 6.5 8.0 7.25 6.0 8.1 7.5 6.5 6.4 7.0 Fundamentos de Programação
tensor(4)

```

Figura 39 - Resultado da recomendação para cada caso conhecido de teste.

Todos os Jupyter *notebooks* usados no desenvolvimento do trabalho, e seus respectivos históricos, estão disponíveis no repositório aberto do GitHub e também estão disponíveis no repositório do GitLab da UFSC, junto com as planilhas usadas e o arquivo do micro serviço em Python.

Repositório do GitHub: https://github.com/maelleal/TCC_Recomendacao_INE

Repositório do GitLab: <https://codigos.ufsc.br/TCC-UFSC/-/tree/tcc-ismael-2022-1>

4.2.7 Design de interface do módulo de recomendação

Nesta seção são apresentadas as novas telas implementadas do módulo de recomendação do sistema de gestão de TCCs do INE e respectivas histórias de usuário implementadas nas novas telas (Quadros 1 a 4).

Quadros 1 - Design de interface US01.

US01 - Temas de projetos definidos pelo professor	
Cadastro do tema	
Lista de temas	
Tema desativado	
Tema excluído	

Quadros 2 - Design de interface US02.

US02 - Visualizar propostas de temas

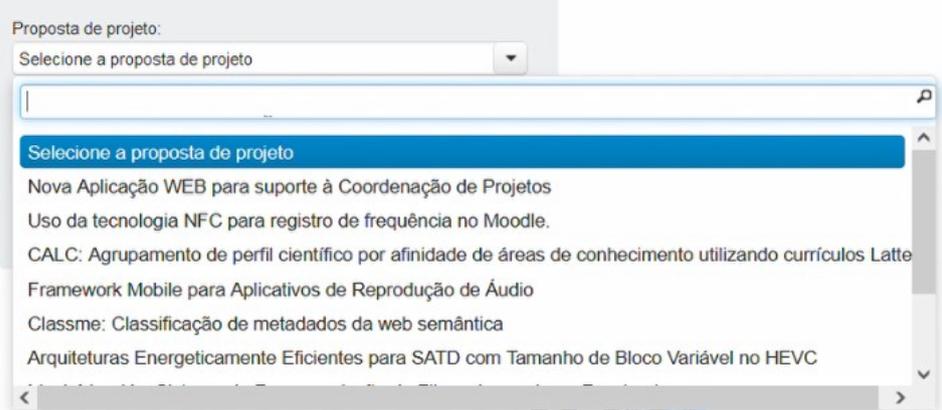
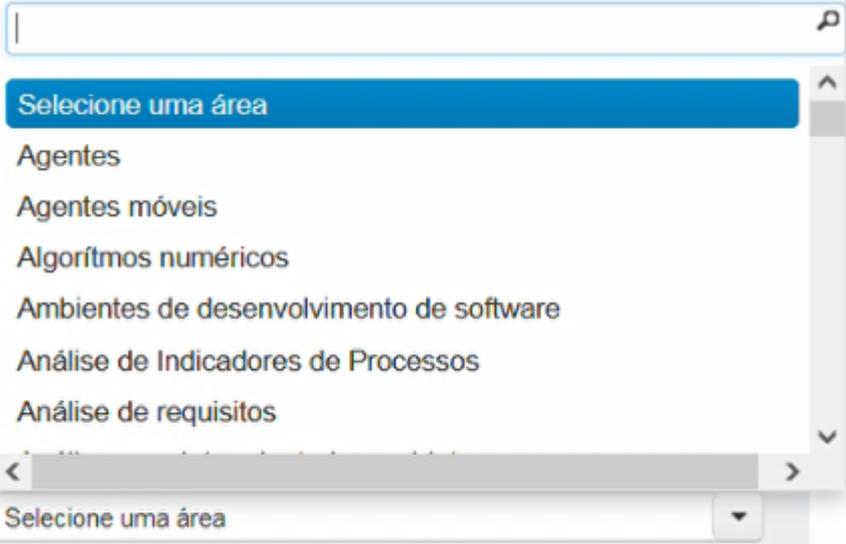
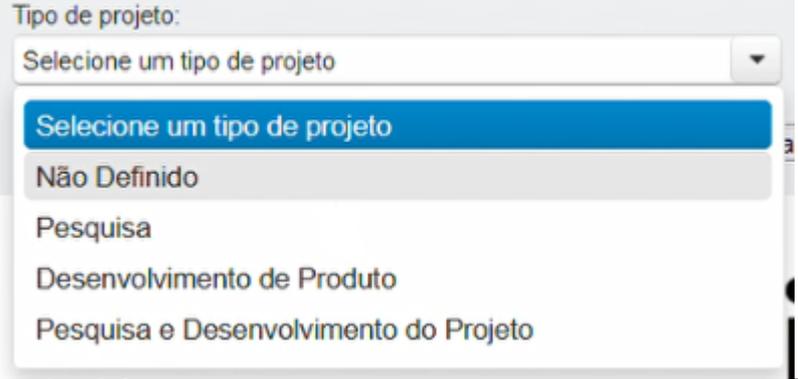


Quadros 3 - Design de interface US03.

<p align="center">US03 - Recomendação de área e propostas</p>				
<p>Opção de recomendação</p>	<p align="center">Recomendação de temas de acordo com meu histórico escolar</p> <div align="center" style="border: 1px solid gray; padding: 5px; margin: 10px auto; width: 80%;"> Recomendação personalizada. </div>			
<p>Área recomendada</p>	<p>Área recomendada</p> <p align="center">Engenharia de Software, Gerência de Projetos</p> <p>Professores e temas:</p> <p align="center">Professores com interesse na área recomendada: Christiane Anneliese Gresse Von Wangenheim - Frank Augusto Siqueira - Joao Candido Lima Dovicchi - Mauricio Floriano Galimberti - Raul Sidnei Wazlawick - Ricardo Pereira e Silva - Fabiane Barreto Vavassori Benitti - Jean Carlo Rossa Hauck - Jose Eduardo de Lucca - Patricia Vilain -</p>			
<p>Propostas com a área recomendada</p>	<p>Temas de pesquisa abertos com a área recomendada.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%; padding: 5px;"> </td> <td style="width: 33%; padding: 5px;"> </td> <td style="width: 33%; padding: 5px;"> </td> </tr> </table>			

Quadros 4 - Design de interface US04.

<p>US04 - Cadastrar TCC</p>

Lista de propostas	
Lista de áreas	
Lista de tipos de projeto	
TCC cadastrado com os vínculos de proposta, área e tipo	<p>Informações</p> <p>TCC Parceiro Responsável Orientação Coorientação Banca</p> <p>Título: Meu TCC Descrição: Meu TCC Data de Criação: 16/11/2022 Apto para defender: Não Proposta vinculada: Minha proposta de projeto Teste Tipo projeto: Pesquisa Área projeto: Algoritmos numéricos Status: Em Andamento</p>

5 AVALIAÇÃO DO MÓDULO DE RECOMENDAÇÃO

Com o objetivo de avaliar o módulo de recomendação proposto foram realizadas duas avaliações. Uma avaliação do modelo de recomendação baseado em *Machine Learning*, já descrita na sessão 4.2.6 deste trabalho, e outra avaliação em relação à utilidade, desempenho, funcionalidade e usabilidade pela perspectiva de alunos que estão cursando a disciplina de Introdução a Projetos, nosso público-alvo.

Deste modo, foi realizada a avaliação por usuários de forma a obter dados sobre a qualidade percebida pelos alunos.

5.1 AVALIAÇÃO POR USUÁRIOS

Com o objetivo de avaliar o módulo desenvolvido faz-se necessário um teste com usuários em um ambiente controlado.

5.1.1 Definição da avaliação por usuários

A avaliação por usuário é, de maneira simplificada, uma forma sistemática de observar o usuário usando uma ferramenta e assim coletar informações específicas que permitam determinar se a ferramenta foi percebida de maneira fácil, ou não, por eles. Assim são realizados os testes de usabilidade para conseguir informações através dos dados gerados.

A avaliação pode ser separada em duas visões, a do professor e a do aluno. Para a avaliação do professor, foi realizada uma entrevista com o Coordenador de projetos de

conclusão de curso e administrador do Sistema de Gestão de TCCs do INE, logo após a apresentação e demonstração de uso do novo módulo. Com a entrevista, foram anotados pontos de melhorias e dúvidas pertinentes para eventuais correções ou melhorias do sistema. Já para os alunos, foi aplicado um questionário avaliando utilidade, desempenho, funcionalidade e a usabilidade, após o uso do módulo. Os fatores de qualidade avaliados, com os alunos, são apresentados na Tabela 6.

Tabela 6 - Questionário de satisfação.

Característica	Questionário do Aluno
Utilidade	O módulo de recomendação é útil na escolha de um tema de TCC.
	Eu possivelmente aceitaria realizar o meu TCC com um dos temas de pesquisa indicados pelo módulo de recomendação.
Funcionalidade	Eu achei a recomendação de área para tema de TCC coerente.
	As informações disponibilizadas como resultado da recomendação são suficientes.
	Informe se você observou algum erro (bug) em relação à funcionalidade do módulo.
Desempenho	A tarefa de verificar a área recomendada foi concluída?
	Quanto tempo aproximadamente você levou para cumprir a tarefa de verificar a área recomendada?
Satisfação e Usabilidade	Eu penso que usarei este sistema com frequência
	Acho o sistema desnecessariamente complexo
	Penso que o sistema é fácil de usar
	Acho que vou precisar de ajuda de um técnico para usar este sistema
	Acho as funções deste sistema bem integradas
	Encontro muitas inconsistências neste sistema
	Imagino que as pessoas aprenderão rapidamente a usar este sistema
	Acho o sistema pouco prático de usar
	Senti-me confiante ao usar o sistema
	Precisei aprender muitas coisas antes de ser capaz de operar o sistema

Com o auxílio de uma ferramenta de formulários foram coletados os dados pósteste. O questionário foi definido com os fatores apresentados na Tabela 6 e é apresentado no Anexo A. No teste realizado com os alunos, os usuários receberam, primeiramente, uma visão geral do objetivo do módulo e o que deveriam executar como tarefa pré-definida (realizar a recomendação de área de acordo com o seu histórico escolar).

5.1.2 Execução da avaliação por usuários

A execução da avaliação do módulo de recomendação foi realizada por alunos do INE cursando a disciplina de Introdução a Projetos, que precede a elaboração do TCC, momento no qual tipicamente os alunos estão selecionando um tema de TCC, e pelo professor coordenador de TCCs do INE.

No total foram realizadas 11 avaliações, sendo uma com um professor que é a pessoa responsável pela administração do Sistema de Gestão de TCC do INE e as demais 10 avaliações foram de alunos potenciais usuários do módulo. A avaliação realizada pelos alunos foi acompanhada pelo autor do trabalho, utilizando seu próprio computador de desenvolvimento para a execução do sistema Gestão de TCCs em ambiente de homologação.

5.1.3 Análise dos dados

Para uma melhor interpretação das informações, as respostas são agrupadas por critério de análise, apresentando a quantidade referente a respostas.

Sobre a utilidade:

Os usuários consideram que o módulo de recomendação é útil para os alunos e que, em sua maioria, concordam com a possibilidade de usar um tema de pesquisa indicado na recomendação. De forma geral, os alunos disseram que um direcionamento de área ou indicação de proposta de projeto ajuda muito e elogiaram a ideia e a proposta do trabalho. Os dados coletados sobre a utilidade são apresentados na Tabela 7.

Tabela 7 - Respostas sobre usabilidade

Sobre a utilidade	Concordo totalmente	Concordo	Nem discordo nem concordo
O módulo de recomendação é útil na escolha de um tema de TCC.	9 alunos	1 aluno	
Eu possivelmente aceitaria realizar o meu TCC com um dos temas de pesquisa indicados pelo módulo de recomendação.	2 alunos	6 alunos	2 alunos

Sobre o desempenho:

A parte de avaliação de desempenho do módulo de recomendação não foi totalmente satisfatória devido a dificuldades enfrentadas com o ambiente de homologação do sistema. Por conta do *login* centralizado da UFSC, não foi permitido que os alunos acessassem o sistema, que estava em execução no computador de desenvolvimento, pelos computadores do laboratório onde as avaliações foram realizadas. Outra dificuldade encontrada foi em relação aos dados de cada aluno na base de dados de homologação. Como a base de dados de homologação não é atualizada com os dados do semestre corrente, boa parte dos alunos não estavam cadastrados no sistema de Gestão de TCCs do INE, não sendo possível realizar a busca de histórico escolar no CAGR. Dois dos alunos já estavam cadastrados na base de homologação e para outros dois alunos foi inserido o seu número de matrícula diretamente no código, para que fosse possível buscar o histórico no CAGR. Com esses alunos foi possível observar o resultado da recomendação com os seus respectivos históricos. Os demais alunos observaram o comportamento e o resultado do módulo de recomendação com os dados de históricos de autor do trabalho. Assim, foi necessário que os alunos participantes tivessem auxílio para terminar a tarefa e, em sua maioria, eles relataram que demoraram entre 2 e 4 minutos para concluir a tarefa de executar a recomendação. Na Tabela 8 podemos observar os dados coletados no questionário.

Tabela 8 - Respostas sobre desempenho

Sobre o desempenho	De 0 a 2 minutos	De 2 a 4 minutos
Quanto tempo aproximadamente você levou para cumprir a tarefa de verificar a área recomendada?	2 alunos	8 alunos
A tarefa de verificar a área recomendada foi concluída?	Todos responderam Sim	

Sobre a funcionalidade:

Em relação à funcionalidade, o sistema de recomendação teve um *feedback* positivo. Os alunos, em sua maioria, tiveram o entendimento de que a área recomendada e as informações disponíveis, resultantes da recomendação, são suficientes para auxiliá-los. Os alunos alegaram não encontrar erros no sistema, bem como o professor entrevistado. O professor comentou sobre a possibilidade de haver uma especificação de área, ou um subgrupo da área, no *card* de proposta de projeto. Os dados obtidos sobre a funcionalidade no questionário são apresentados na Tabela 9.

Tabela 9 - Respostas sobre funcionalidade

Sobre a funcionalidade	Concordo totalmente	Concordo	Nem discordo nem concordo
Eu achei a recomendação de área para tema de TCC coerente.	6 alunos	4 alunos	
As informações disponibilizadas como resultado da recomendação são suficientes.	5 alunos	3 alunos	2 alunos
Informe se você observou algum erro (<i>bug</i>) em relação à funcionalidade do módulo.	Todos responderam que não encontraram <i>bug</i>		

Sobre a usabilidade:

Quanto à usabilidade, pode-se observar um *feedback* positivo na facilidade de uso perante os alunos. Foi notado um tempo de observação dos *cards* apresentados antes de clicarem no botão para realização da recomendação personalizada, mas o que não caracteriza uma dificuldade na utilização.

Em termos de satisfação, a avaliação de usabilidade foi realizada por meio da abordagem de avaliação de usabilidade SUS (BROOKE, 1996) e observou-se uma pontuação média acima de 90 pontos de usabilidade, sendo uma pontuação satisfatória obtida. A Tabela 10 mostra os valores SUS coletados no questionário.

Tabela 10 - Pontuação SUS

	Pontuação SUS
Aluno 1	87,50
Aluno 2	92,50
Aluno 3	85,10
Aluno 4	85,00
Aluno 5	100,00
Aluno 6	97,50
Aluno 7	92,50
Aluno 8	90,00
Aluno 9	95,70
Aluno 10	92,50
Média	91,83

Pontos fortes

Com base nos *feedbacks* dos usuários, verbais ou por meio do questionário, o módulo de recomendação do Sistema de Gestão de TCCs do INE foi considerado útil, funcional, com bom desempenho e com boa usabilidade. Foram observados como pontos fortes a listagem de professores vinculados à área recomendada, bem como a exibição de *cards* com propostas de projetos, daquela área recomendada, disponibilizadas pelos professores. Também foi tido como ponto forte, por parte do professor entrevistado, o vínculo de um TCC com uma área e um tipo de projeto, o que será útil para aprimorar o modelo de recomendação com o passar do tempo.

Sugestões de melhoria

Como pontos de melhoria foi sugerido que o sistema de recomendação pudesse realizar a inferência sem o histórico escolar, apenas com um possível orientador e/ou com uma disciplina que o aluno possui muita afinidade. Além disso, foi sugerido que o nome dos professores que não exercem a função de orientador ou não estão mais no quadro de professores do departamento, não sejam exibidos na lista de professores vinculados com a área. Também foi sugerido que seja permitida a inserção de uma especificação, ou subárea, da área no *card* de proposta de projeto, bem como colocar um “*hint*” das disciplinas que fazem parte daquela área, quando o *mouse* estiver sobre o campo de área.

6 CONCLUSÃO

O presente trabalho apresenta o desenvolvimento de um módulo de recomendação de temas e projetos baseado em *Machine Learning* para o Sistema de Gestão de TCCs do INE. Para o desenvolvimento do trabalho foi inicialmente realizado o estudo da literatura, levantamento do estado da arte, a coleta e preparação do *dataset*, treinamento do modelo de IA e o desenvolvimento do módulo.

Para a análise do estado da arte foi realizado um mapeamento sistemático da literatura, por meio do qual foram encontrados três trabalhos relacionados, de acordo com os critérios de inclusão e exclusão. Em um dos trabalhos é realizado o estudo de um sistema de recomendação que visa validar a compatibilidade entre orientando e orientador, tendo como base o histórico escolar do orientando e um questionário respondido pelos orientadores. Em um outro trabalho são apresentados resumos, muito bem elaborados, sobre alguns dos principais modelos de aprendizado de máquina utilizados em sistemas de recomendação, contando com um *dataset* com dados tabulares como *input* e, concluindo, que um sistema de filtro colaborativo parece mais promissor que os demais algoritmos estudados. Já o terceiro trabalho estuda a questão de interpretabilidade de sistemas de recomendação, baseada em *deep learning*, e também traz o estudo de diversos sistemas de recomendação, mostrando as vantagens e desvantagens dos modelos.

Foi então realizado o levantamento e análise dos requisitos, por meio de entrevistas não estruturadas. Os requisitos do trabalho foram analisados e documentados na forma de histórias de usuário e estão definidos na sessão 4.1.

Para possibilitar o desenvolvimento e treinamento de um modelo de recomendação, baseado em Inteligência Artificial, os dados, que estavam em forma não estruturada, em dois diferentes sistemas, foram extraídos e tratados para possibilitar um bom resultado. Foram realizados diversos testes com diferentes tipos de modelos e bibliotecas, além de diferentes estruturas de dados para alimentar o modelo. Ao final, foi utilizado um modelo de classificação tabular com a biblioteca FasAi. Pode-se notar que os valores de validação do sistema de recomendação foram satisfatórios e que o modelo obteve um comportamento melhor quando o

conjunto de dados possuía orientador e/ou disciplina na tupla do TCC em questão. O modelo de recomendação foi exportado para ser utilizado em conjunto com o Sistema de Gestão de TCCs do INE.

A implementação de um módulo de recomendação de temas de TCC para o sistema Gerenciamento de TCCs do INE passou por vários pontos da estrutura do sistema, tendo desde criação e alteração de tabelas da base de dados, criação e alteração de *view*, *controllers* e *models*, até a criação de um subsistema em Python que, ao receber os dados do aluno, executa o modelo de recomendação, com esses dados como *input*, e tem como resposta uma área que será passada para o sistema e exibida em tela, além de ser usada como filtro para listagem de professores e proposta de projetos.

Uma avaliação inicial do módulo desenvolvido levantou indícios de que as informações resultantes da recomendação são satisfatórias e suficientes para auxiliar os alunos que estão, de certa forma, perdidos em relação a qual assunto estudar no seu trabalho de conclusão de curso. Também foi constatado que os *cards* de proposta de projetos exibidos, sejam eles filtrados pela área recomendada ou não, vão ajudar tanto os professores a conseguirem alunos para seus projetos de pesquisa quanto os alunos, que estão em busca de alguma proposta. A avaliação também visou saber sobre a usabilidade, funcionalidade e desempenho do módulo e, de acordo com os dados obtidos nas respostas do questionário, verificou-se um resultado satisfatório por parte dos alunos.

Como trabalhos futuros sugere-se, primeiramente, a implantação do novo módulo no ambiente de produção do Sistema de Gestão de TCCs do INE, para que os alunos e professores possam usufruir dos benefícios do módulo. Com isso, é importante que seja feita uma nova avaliação, agora em ambiente de produção. Também é recomendado que seja implementado um retreinamento automatizado do modelo de recomendação a cada semestre, ou seja, com o término de um semestre algum gatilho acionaria uma extração de dados estruturados da base de dados, passando pelo código de treinamento do modelo de recomendação e exportando o modelo para substituir o atual. Assim, a cada semestre o modelo de recomendação seria aprimorado. Também pode-se pensar em aprimorar o sistema de recomendação usando além da classificação tabular, algum tipo de ontologia, ou até mesmo tendo mais fontes de dados para servir de *input* para o modelo, como os dados dos questionários de avaliação de disciplinas e professores que ocorre aos finais de cada semestre, ou até mesmo de algum questionário, dentro do próprio Sistema de Gestão de TCCs do INE, para avaliar a compatibilidade entre o aluno e

professores orientadores de determinada área. Outra importante ideia sugerida para trabalhos futuros seria o uso de algoritmo para realizar a classificação de TCCs de acordo com palavras-chave encontradas no texto, agrupando os trabalhos em diferentes áreas.

REFERÊNCIAS

- BAUER, Christian; GREGORY, Gary. **Java Persistence with Hibernate**. Estados Unidos, Manning Publications, 2015.
- BERGSTEN, Hans. **JavaServer Faces**. Estados Unidos, O'Reilly Media, 2009.
- BONILLA, Camila Jardim Cavalcante. **Uma introdução aos sistemas de recomendação: modelos matemáticos, algoritmos e aplicações**. 2020. 1 recurso online (118 p.) Dissertação (mestrado profissional) - Universidade Estadual de Campinas, Instituto de Matemática Estatística e Computação Científica, Campinas, SP.
- BOTELHO, Felipe Gonçalves; UGIONI, Pedro Henrique Rocha. **TCC UFSC - Nova Aplicação WEB para suporte à Coordenação de Projetos**. 2015. 67 f. TCC (Graduação) - Curso de Ciências da Computação, Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis, 2015.
- BRASCHER, Marisa; CAFÉ, Ligia. Organização da informação ou organização do conhecimento? In: ENCONTRO NACIONAL DE PESQUISA EM CIÊNCIA DA INFORMAÇÃO, 9., 2008, São Paulo. Anais... São Paulo: ENANCIB, 2008. p. 7.
- BROOKE, John. "SUS-A quick and dirty usability scale." Usability evaluation in industry, 1996. Disponível em: <<https://www.crepress.com/product/isbn/9780748404605>>.
- COLEGIADO DO CURSO DE SISTEMAS DE INFORMAÇÃO. **Regimento interno para elaboração de trabalho de conclusão de curso (TCC) de Sistemas de Informação**. Disponível em: <https://sin.paginas.ufsc.br/files/2011/09/RI-TCC-SIN_v15marco.pdf>. Acesso em: abril de 2021.
- DUBOIS, Paul. **MySQL**, Developer's library. Reino Unido, Addison-Wesley, 2013.
- ELLIOTT, James; O'BRIEN, Timothy.M.; FOWLER, Ryan. **Harnessing Hibernate**. Estados Unidos, O'Reilly Media, 2008.
- ERRICA, Davide Bacciu Federico; MICHELI, Alessio; PODDA, Marco. A gentle introduction do deep learning for graphs. v. 46, n. 1, p. 5, 2020.
- ESCOVEDO, Tatiana. **Machine Learning: Conceitos e Modelos — Parte II: Aprendizado Não-Supervisionado***, 2020. Disponível em: <<https://tatianaesc.medium.com/machine-learning-conceitos-e-modelos-parte-ii-aprendizado-n%C3%A3o-supervisionado-fb6d83e4a520>>. Acesso: setembro 2021.
- ESCOVEDO, Tatiana. **Machine Learning: Conceitos e Modelos — Parte I: Aprendizado Supervisionado***, 2020. Disponível em: <<https://tatianaesc.medium.com/machine-learning-conceitos-e-modelos-f0373bf4f445>>. Acesso: setembro 2021.

FASTAI, **Repositório GitHub**, 2019. Disponível em: < <https://github.com/fastai/fastai/> > Acesso: agosto 2021.

FEITOSA, Douglas & Silva, Vitor & Lopes, Roberta & Paraguaçu, Fábio. (2011). **Recomendação Assistida por Computador para Orientação de Trabalhos de Conclusão de Curso: Baseada no Perfil Informacional e na Capacidade de Orientação Docente.** Informação & Sociedade: Estudos. 21. 131-143.

FIOCRUZ, **Google Colab**, 2021. Disponível em: < <https://bigdata.icict.fiocruz.br/google-colab> >. Acesso: setembro 2021.

FOWLER, Martin. **Injeção de dependência.** Disponível em: < <https://archive.is/ALSVa#selection-1899.1-1899.21>>. Acesso em: julho 2021.

GAVRILOVA, Yulia. **Artificial Intelligence vs. Machine Learning vs. Deep Learning: Essentials.** Disponível em: <<https://ai.plainenglish.io/data-science-vs-artificial-intelligence-vs-machine-learning-vs-deep-learning-50d3718d51e5>>. Acesso: setembro 2021.

GONÇALVES, Diego Fretta. **Melhoria da usabilidade do sistema de tcc do ine/ufsc do ponto de vista do aluno.** 2016. 117 f. TCC (Graduação) - Curso de Ciências da Computação, Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis, Sc, 2016.

GOOGLE, **O que é o Colaboratory?**, 2021. Disponível em: < <https://colab.research.google.com/notebooks/welcome.ipynb?hl=pt-BR> >. Acesso: setembro 2021.

HOELLER, Juergen; JOHNSON, Rod. **Expert One-on-One J2EE Development Without EJB.** Alemanha, Wiley, 2004.

JUPYTER, **The Jupyter Notebook**, 2021. Disponível em: < <https://jupyter.org/> >. Acesso: setembro 2021.

KITCHENHAM, B. **Guidelines for performing Systematic Literature Reviews in Software Engineering** - Version 2.3. EBSE Technical Report, Keele University and University of Durham, 2007.

KOFLER, Michael. **The Definitive Guide to MySQL**, Second Edition. Nova York, Apress, 2004.

KOSHIYAMA, Adriano., ESCOVEDO, Tatiana. **Introdução a Data Science: Algoritmos de Machine Learning e métodos de análise.** Brasil: Casa do Código, 2020.

LEON, Gabriela Vieira. Agrupamento de grafos e sistema de recomendação: um estudo de caso das avaliações da Amazon. 2021.

MISRA, R.; WAN, M.; MCAULEY, J. Decomposing fit semantics for product size recommendation in metric spaces. RecSys '18: Proceedings of the 12th ACM Conference on Recommender Systems, 2018. Citado na página 104.

MORANDINO, Matheus; RODRIGUES, Michelly. **Algoritmos de recomendação: mais presente no dia a dia do que você pensa**, 2020. Disponível em: <<http://www.each.usp.br/petsi/jornal/?p=2684>>. Acesso: setembro 2021.

ONGHERO, Eduardo Demeneck. **Implementação de Melhorias no Sistema de TCCs do Departamento de Informática e Estatística da UFSC: Perfil Professor**. 2018. Disponível em: <<https://repositorio.ufsc.br/handle/123456789/192165>>. Acesso: setembro 2021.

PATEL, Ashish. **Machine Learning Algorithm Overview**, 2018. Disponível em: <<https://medium.com/ml-research-lab/machine-learning-algorithm-overview-5816a2e6303>>. Acesso: setembro 2021.

PIO, Bruno Luiz de Assis. **Uma Introdução às Redes Neurais Artificiais para Estudos em Ecologia**. Brasil, Creative Commons, 2009.

RAY, Tieman. **Fast.ai's software could radically democratize AI**, 2018. Disponível em: <<https://www.zdnet.com/article/fast-ais-new-software-could-radically-democratize-ai/>>. Acesso: setembro 2021.

REIS, Linda G. **Produção de monografia da teoria à prática o método educar pela pesquisa (mep)**. 2. ed. Brasília: SENAC, 2008.

ROBINS. Mark. **The Difference Between Artificial Intelligence, Machine Learning and Deep Learning**, 2020. Disponível em: <<https://www.intel.com.br/content/www/br/pt/artificial-intelligence/posts/difference-between-ai-machine-learning-deep-learning.html>>. Acesso em setembro de 2021.

RUSSELL, STUART J. (STUART JONATHAN), 1962- **Inteligência artificial** / Stuart Russell, Peter Norvig; tradução Regina Célia Simille. – Rio de Janeiro: Elsevier, 2013.

SALLES, A. and Willrich, R. (2015). Recommending web service based on ontologies for digital repositories. In Proceedings of the 21st Brazilian Symposium on Multimedia and the Web, pages 65–72. ACM.

SAMUEL, Arthur L. **Machine learning**. *The Technology Review*, v. 62, n. 1, 1959.

SANTANA. Marlesson. **Deep Learning para Sistemas de Recomendação (Parte 1) — Introdução**. Disponível em: <<https://medium.com/data-hackers/deep-learning-para-sistemas-de-recomenda%C3%A7%C3%A3o-parte-1-introdu%C3%A7%C3%A3o-b19a896c471e>>. Acesso em: setembro 2021.

SNELL, James., TIDWELL, Doug., KULCHENKO, Pavel. **Programming Web Services with SOAP**. Estados Unidos, O'Reilly & Assoc Inc, 2001.

SIMON, Phil. **Too Big to Ignore: The Business Case for Big Data**. Reino Unido: Wiley, 2015

STEFANINI. Stefanini Group. **Machine Learning × Deep Learning: entenda a diferença**, 2019. Disponível em: <<https://stefanini.com/pt-br/trends/artigos/machine-learning-vs-deep-learning>> . Acesso em: maio 2021.

TECHVIDVAN. **Reinforcement Learning Algorithms and Applications**, 2020. Disponível em: <<https://techvidvan.com/tutorials/reinforcement-learning/>>. Acesso: setembro 2021.

THAKUR, Naresh. **The differences between Data Science, Artificial Intelligence, Machine Learning, and Deep Learning**, 2020. Disponível em: <<https://ai.plainenglish.io/data-science-vs-artificial-intelligence-vs-machine-learning-vs-deep-learning-50d3718d51e5>> . Acesso: setembro 2021.

TRINDADE, Joana Filipa Vieira. **Interpretabilidade em Modelos de Sistemas de Recomendação**. 2020. 1 recurso online (165 p.) Dissertação (mestrado profissional) - Universidade do Porto, Departamento de Ciências e Computadores, Porto, Portugal.

VAN VEEN, Fjodor. **The Neural Network Zoo**, 2019. Disponível em: <<https://www.asimovinstitute.org/neural-network-zoo/>>. Acesso: setembro 2021.

WAZLAWICK, Raul Sidnei. **Metodologia de Pesquisa Para Ciência da Computação**. Rio de Janeiro: Elsevier, 2009.

WEISSMANN, Henrique Lobo. **Vire o jogo com Spring Framework**. Brasil, Casa do Código, 2014

WINDER Ph.D., Phil. **Reinforcement Learning**. Estados Unidos: O'Reilly Media, 2020.

Anexo A – Questionário de avaliação - Módulo de recomendação do Sistema de TCCs do INE (Alunos)

Logo Este questionário tem como objetivo avaliar a usabilidade e utilidade do módulo de recomendação de temas do Sistema de TCCs do INE. Esta pesquisa faz parte do Trabalho de Conclusão de Curso de Sistemas de Informação da UFSC de Ismael Aguirre Leal.

Termo de Consentimento Livre e Esclarecido

Você está sendo convidado a participar de um estudo de avaliação do módulo de recomendação do Sistema de TCCs do INE, realizado pelo aluno Ismael Aguirre Leal, do curso de Sistemas de informação da UFSC, com o objetivo de avaliar a usabilidade e utilidade do módulo proposto e suas implicações.

A sua participação voluntária e gratuita no referido estudo consiste em responder o questionário de avaliação do uso da ferramenta.

Você confirma que recebeu esclarecimentos sobre a pesquisa e está ciente de que sua privacidade será respeitada, pois este questionário não armazena qualquer informação que possa identificá-lo. Todos os dados são coletados de forma anônima. Você confirma que foi informado de que pode se recusar a participar do estudo, ou retirar seu consentimento a qualquer momento, sem precisar justificar. É assegurada a assistência durante toda a pesquisa. Ao concordar com este termo, você confirma o seu livre consentimento em participar.

Qualquer dúvida, enviar e-mail para: ismael.leal@grad.ufsc.br

Termos e condições

- Concordo com o Termo de Consentimento

Sobre a utilidade:

O módulo de recomendação é útil na escolha de um tema de TCC. *

Discordo Totalmente

- 1
- 2
- 3
- 4
- 5

Concordo Totalmente

Eu possivelmente aceitaria realizar o meu TCC com um dos temas de pesquisa indicados pelo módulo de recomendação. *

Discordo Totalmente

- 1
- 2
- 3
- 4
- 5

Concordo Totalmente

Sobre a funcionalidade:

Eu achei a recomendação de área para tema de TCC coerente. *

Discordo Totalmente

- 1
- 2
- 3
- 4
- 5

Concordo Totalmente

As informações disponibilizadas como resultado da recomendação são suficientes. *

Discordo Totalmente

- 1
- 2
- 3
- 4
- 5

Concordo Totalmente

Informe se você observou algum erro (bug) em relação à funcionalidade do módulo. *

Sobre a desempenho:

A tarefa de verificar a área recomendada foi concluída? *

- Sim
- Não

Quanto tempo aproximadamente você levou para cumprir a tarefa de verificar a área recomendada? *

- De 0 a 2 minutos
- De 2 a 4 minutos
- De 4 a 6 minutos
- Mais de 6 minutos

Sobre a satisfação de usabilidade:

Avalie a usabilidade do módulo *

	Discordo totalmente	Discordo parcialmente	Nem discordo nem concordo	Concordo	Concordo Totalmente
Eu penso que usarei este sistema com frequência	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Acho o sistema desnecessariamente complexo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Penso que o sistema é fácil de usar	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Acho que vou precisar de ajuda de um técnico para usar este sistema	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Acho as funções deste sistema bem integradas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Encontro muitas inconsistências neste sistema	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Imagino que as pessoas aprenderão rapidamente a usar este sistema	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Acho o sistema pouco prático de usar	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Senti-me confiante ao usar o sistema	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Precisei aprender muitas coisas antes de ser capaz de operar o sistema	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Um módulo de recomendação de projetos baseado em *machine learning* para o Sistema de Gestão de TCCs do INE

Ismael Aguirre Leal

Departamento de Informática e Estatística
Universidade Federal de Santa Catarina (UFSC) – Florianópolis – SC – Brasil

Ismael.leal@grad.ufsc.br

Abstract. *Currently, the INE/CTC/UFSC TCCs Management System provides students with a quick search for possible advisors and their areas of interest. Despite this possibility, the proposed topics are usually presented in a generic and comprehensive way. A more detailed presentation of themes by professors, as well as the possibility of a theme or area recommendation for students, according to the possible aptitudes and affinities presented during the undergraduate course, could facilitate the choice of theme for the Final Paper and possibly improve the result at the end of the TCC. Thus, this work presents the development of a project recommendation module based on Machine Learning for the INE/CTC/UFSC TCCs management system*

Resumo. *Atualmente o Sistema de Gestão de TCCs do INE/CTC/UFSC proporciona aos alunos uma busca rápida de possíveis orientadores e suas áreas de interesse. Apesar dessa possibilidade, geralmente os temas propostos são apresentados de forma genérica e abrangente. A apresentação mais detalhada de temas por parte dos professores, bem como a possibilidade de uma recomendação de tema ou área, para os alunos, de acordo com as possíveis aptidões e afinidades apresentadas durante o curso de graduação, poderiam facilitar a escolha do tema do TCC e possivelmente melhorar o resultado ao final do TCC. Assim, este trabalho apresenta o desenvolvimento de um módulo de recomendação de projetos baseado em Machine Learning para o sistema de gerenciamento de TCCs do INE/CTC/UFSC.*

1. Introdução

É muito comum que alunos de cursos de graduação cheguem às últimas fases dos seus cursos sem ideia alguma sobre qual tema escolher para o seu Trabalho de Conclusão de Curso (TCC). A escolha de um objetivo de pesquisa é frequentemente a tarefa mais difícil em um trabalho de mestrado ou doutorado (WAZLAWICK, 2009), o que também costuma valer para um trabalho de conclusão de curso. Então, alunos iniciando seus TCCs acabam percebendo que possuem pouco tempo para buscar um tema de seu interesse e que combine com um orientador de sua afinidade. Ainda segundo Wazlawick (2009), não se recomenda uma pesquisa cujo tema não seja compatível com os conhecimentos do orientador. Somente depois da escolha do tema, do objetivo e do orientador é que o aluno começará o seu trabalho (SIN, 2011).

O Departamento de Informática e Estatística (INE) da Universidade Federal de Santa Catarina (UFSC), que oferece os cursos de Sistemas de Informação (SIN) e Ciências da Computação (CCO), possui um software para gerenciamento de TCCs do departamento. Trata-se de um Sistema de Gestão de Trabalhos de Conclusão de Curso que permite a entrega de documentos e acompanhamento de calendário das disciplinas vinculadas ao TCC, por parte do aluno. Também permite aos orientadores e membros de bancas acompanharem e avaliarem os projetos (BOTELHO; UGIONI, 2015).

Atualmente, o Sistema de Gestão de TCC do INE permite visualizar uma lista de professores e suas respectivas áreas de pesquisa, nas quais se interessam por orientar trabalhos. Porém o sistema carece de um ambiente que fomente ideias de temas de pesquisa mais específicos para os alunos. Segundo Reis (2008), o tema a ser escolhido depende de vários aspectos, como o interesse do aluno, a experiência pessoal, e, portanto, para tal escolha é vital ao aluno pensar sobre o assunto. Quanto maior o leque de ideias e temas, maior tende a ser a chance de o aluno encontrar algo que seja de seu interesse. Assim, o desempenho dos alunos nas diversas disciplinas, no decorrer do curso, poderia ser utilizado para indicar potenciais temas de interesse.

2. Metodologia

A metodologia de pesquisa utilizada, visando alcançar os objetivos do presente trabalho, é dividida de tal forma que na primeira etapa está a parte de fundamentação teórica, onde será realizado o estudo e análise de leitura sobre os principais conceitos e teorias a serem abordados ao longo do trabalho. Como próxima etapa temos o desenvolvimento do módulo de recomendação, que diz respeito a implementação do sistema de recomendação com inteligência artificial, junto com as modificações necessárias no Sistema de Gestão de TCC do INE para contemplar o novo módulo. Nesta mesma etapa temos o desenvolvimento de um modelo de inteligência artificial que fará a recomendação de temas e áreas no novo módulo do Sistema de Gestão de TCC do INE. E por último temos a avaliação de resultados, onde é avaliado a qualidade da inteligência artificial do sistema de recomendação, bem como a usabilidade e funcionalidade das modificações realizadas no presente trabalho.

3. Fundamentação teórica

O INE, Departamento de Informática e Estatística da Universidade Federal de Santa Catarina, conta com um sistema de gerenciamentos de projetos de conclusão de curso (TCCs), sendo utilizado tanto por alunos quanto por professores para realizar as mais variadas atividades relacionadas à gestão de TCCs. O sistema foi concebido e iniciado por meio de um trabalho de conclusão de curso, onde os alunos desenvolveram o sistema (BOTELHO; UGIONI, 2013). O Sistema de Gestão de TCC do INE vem sendo desenvolvido quase que de forma colaborativa, tendo funcionalidades incorporadas e melhorias realizadas por alunos em seus trabalhos de conclusão de curso. Um desses trabalhos, (GONÇALVES, 2016) elaborou melhorias significativas na usabilidade do sistema, focando principalmente nos usuários com perfil de aluno. Outros dois trabalhos trouxeram melhorias ao sistema, tendo o aluno (ONGHERO, 2018) realizado melhorias de funcionalidades, como permitir a definição de área de conhecimento, permitir a visualização de TCC em que é membro, visualização de membro e avaliações, além de melhorias de desempenho.

Um sistema de recomendação (*RecSys*), em uma abordagem dita como clássica, é uma aplicação de aprendizado de máquina para negócios que, por meio de modelagem de dados e

aplicação de algoritmos, tenta prever a nota (rating) ou preferência de um usuário a um determinado item. Assim, tais sistemas são classificados de acordo com o tipo de algoritmo usado (MORANDINO; RODRIGUES, 2020). Os *RecSys* podem ser utilizados para recomendar conteúdo em diferentes domínios, como livros, música, filme, varejo, notícias etc. Assim como a visão computacional e processamento de linguagem natural foram radicalmente impactadas por DL, atualmente estamos vivenciando o mesmo na área de Sistemas de Recomendação. Não é só a quantidade de *papers* acadêmicos que estão crescendo e avançando o estado-da-arte nos últimos 4 anos. Empresas grandes estão migrando os sistemas clássicos de recomendação para abordagens com DL (SANTANA, 2018). Ainda existem outros tipos de Sistemas de Recomendação que não se baseiam no modo clássico e sim em vertentes bastante utilizadas, mas não tão difundidas na maioria dos sites e trabalhos que falam sobre o assunto.

A Inteligência Artificial (IA) é um grande campo que abrange lógica, probabilidade e matemática, além de percepção, raciocínio, aprendizado, ação e ainda, tudo o que se refere à eletrônica, desde dispositivos microeletrônicos até robôs (RUSSEL & NORVIG, 2013). O termo 'Inteligência Artificial' foi cunhado por John McCarthy, um cientista da computação, em 1955. Os mesmos autores, Russel & Norvig (2013), definem, sucintamente, a inteligência artificial como o estudo de agentes inteligentes que recebem percepções e dados do ambiente e executam ações (RUSSEL & NORVIG, 2013). Um agente inteligente que aprenderá caso melhore o seu desempenho em alguma tarefa futura, após fazer observações sobre o mundo. Nesse sentido, pode-se dizer que se trata de um aprendizado de máquina.

O corriqueiramente chamado apenas de Colab é uma abreviatura para o nome Google Colaboratory. O Google Colaboratory é um ambiente de notebooks Jupyter que não requer configuração e é executado na nuvem. Escreva e execute códigos em Python, salve e compartilhe suas análises e acesse poderosos recursos de computação científica, tudo gratuitamente no seu navegador (FIOCRUZ, 2021). Os chamados notebooks do Colab permitem combinar código executável e *rich text* em um só documento, além de imagens, HTML, LaTeX e muito mais. Quando você cria seus próprios notebooks do Colab, eles são armazenados na sua conta do Google Drive. É possível compartilhar os notebooks do Colab facilmente com colegas de trabalho ou amigos e permitir que eles façam comentários ou até editem o documento. Os notebooks do Colab são notebooks do Jupyter hospedados no Colab. O Jupyter Notebook é um aplicativo da web de código aberto que permite criar e compartilhar documentos que contêm código ativo, equações, visualizações e texto narrativo. Os usos incluem: limpeza e transformação de dados, simulação numérica, modelagem estatística, visualização de dados, aprendizado de máquina e muito mais (JUPYTER, 2021).

4. Módulo de recomendação

Neste capítulo será abordado o processo de desenvolvimento e implementação relacionados ao novo módulo e às novas telas, propostas neste trabalho, para o Sistema de Gestão de TCC do INE. Também será apresentada uma seção com as tecnologias utilizadas pelo sistema, bem como outra seção com os passos e particularidades inerentes ao processo de configuração de ambiente de desenvolvimento do sistema.

4.1. Preparação do ambiente

Para que fosse possível realizar a implementação do novo módulo para o Sistema de Gestão de TCC do INE, foi necessário, primeiramente, preparar todo ambiente de desenvolvimento. Com intuito de simplificar e facilitar o desenvolvimento, recomenda-se utilizar um IDE (ambiente integrado de desenvolvimento) que vem sendo utilizado pelos alunos e professores que trabalharam no Sistema de Gestão de TCC do INE, o IntelliJ. O IDE IntelliJ fornece um suporte a grande parte das tecnologias envolvidas e ferramentas utilizadas para o desenvolvimento necessário.

Uma visão macro do processo de configuração pode ser observada na Figura 1 que posteriormente será detalhado.

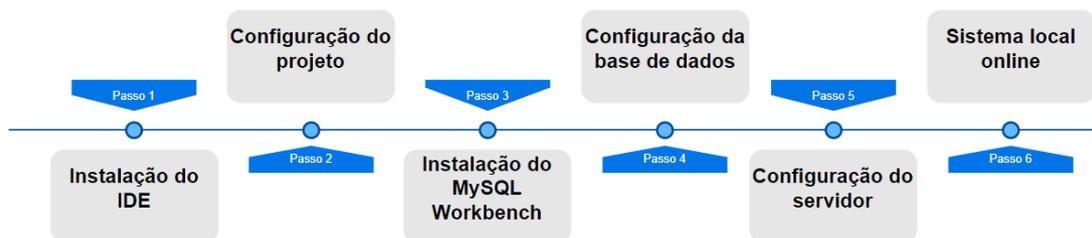


Figura 40 - Fluxograma de configuração de ambiente de desenvolvimento

4.2. Preparação do conjunto de dados

Para que o modelo de recomendação fosse treinado com os dados de notas do histórico escolar dos alunos, foi necessário a extração de tais dados de um segundo sistema, o CAGR. Na base de dados do Sistema de Gestão de TCCs do INE tem-se os dados de “id_ufsc”, número de identificação único para cada aluno vinculado a UFSC, e matrícula dos alunos cadastrados no sistema. De posse desta lista de “ids” o código do Sistema de Gestão de TCCs do INE foi alterado, temporariamente, para buscar as matrículas vinculadas a cada “id”, necessitando filtrar por matrícula de graduação e dos dois cursos estudados no trabalho. Desse modo, criou-se uma nova lista com os dados de matrículas dos alunos onde, para cada matrícula, foi realizada a busca do histórico escolar no CAGR.

Em um primeiro momento os dados de histórico de cada matrícula, foram extraídos de forma bruta, com a tabela contendo todas as notas de todas as disciplinas cursadas por aquele aluno. Como existiam muitos alunos com notas em diferentes disciplinas, de diferentes áreas, sem que houvesse uma homogeneidade das disciplinas cursadas, dado que alguns alunos eram mais antigos e demoraram mais do que o comum para concluírem o TCC, optou-se por transformar os dados. No lugar das notas brutas, foi calculado uma média de notas para cada área de pesquisa, que contém um conjunto de disciplina, de modo que o conjunto de dados, que continha mais de duzentas colunas de notas, uma coluna para cada disciplina, foi reduzido para apenas traze, uma coluna para cada área.

Ainda houve a necessidade da criação de classificação, que foi realizada de forma manual, pelo autor do presente trabalho, para cada um dos TCCs contidos no conjunto de dados, pois tal classificação, até então inexistente, seria usada para treinamento do modelo de recomendação. Cada TCC foi classificado quanto a área do projeto, uma disciplina que mais se aproxima com o trabalho e, posteriormente, do tipo de trabalho.

4.3. Treinamento do modelo de classificação tabular

Para treinar o modelo de recomendação é criado um ambiente de desenvolvimento em nuvem, com o auxílio da ferramenta Google Colab. No ambiente do Colab são inseridos código Python, em um Jupyter notebook, que definirão a lógica de programação responsável pela importação dos dados, classificação das *features*, treinamento do modelo, e testes. Como está sendo utilizada uma biblioteca que facilita a utilização e implementação de *Machine Learning* e, conseqüentemente, de sistema de recomendação, os exemplos disponibilizados pela FastAi nortearam a lógica implementada. Como a estrutura dos dados utilizados é de planilha, ou seja, dados em uma tabela, será utilizado como base um modelo de classificação de dados tabulares, adaptando e alterando o código para a necessidade da estrutura e objetivo do presente trabalho.

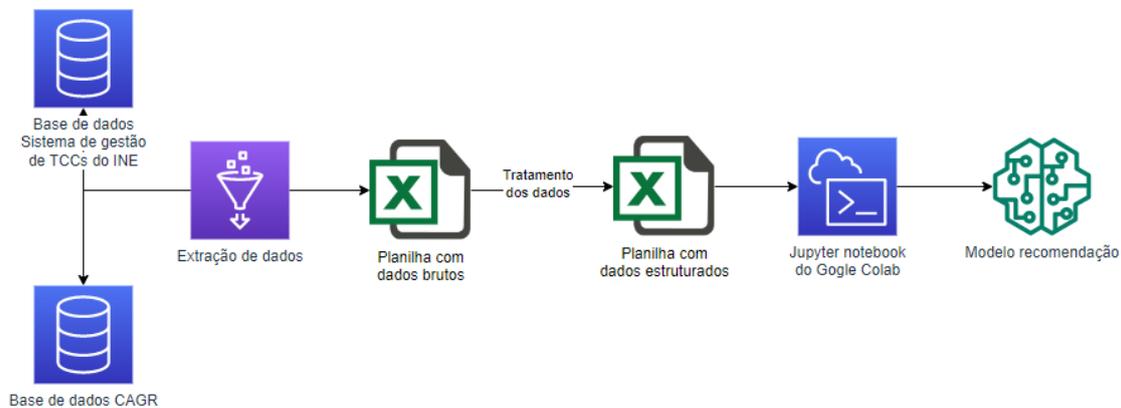


Figura 41 - Arquitetura de desenvolvimento do modelo de recomendação

Para realização da classificação tabular, com FastAi, é usada função que implementa, implicitamente, uma rede neural como modelo de IA. Para o treinamento em questão, foi usado como métrica a acurácia. Vale lembrar que por se tratar de uma divisão aleatória do *dataset*, entre dados para treinamento e dados para validação, sempre que o código é executado obtém-se resultados diferentes para o aprendizado do modelo, mesmo que na maioria das vezes essa diferença seja desprezível. Porém, se nessa divisão do *dataset*, uma das partes ficar com muitos índices, linhas do *dataset*, com dados faltantes para as *features* de “orientador” e “classif_area”, os valores de taxa de aprendizado e o número necessários de épocas para um resultado melhor, pode variar.

Após o treinamento realizado, há a necessidade de validar a classificação realizada pelo modelo treinado. Isso é feito chamando a função de predição em lote da biblioteca.

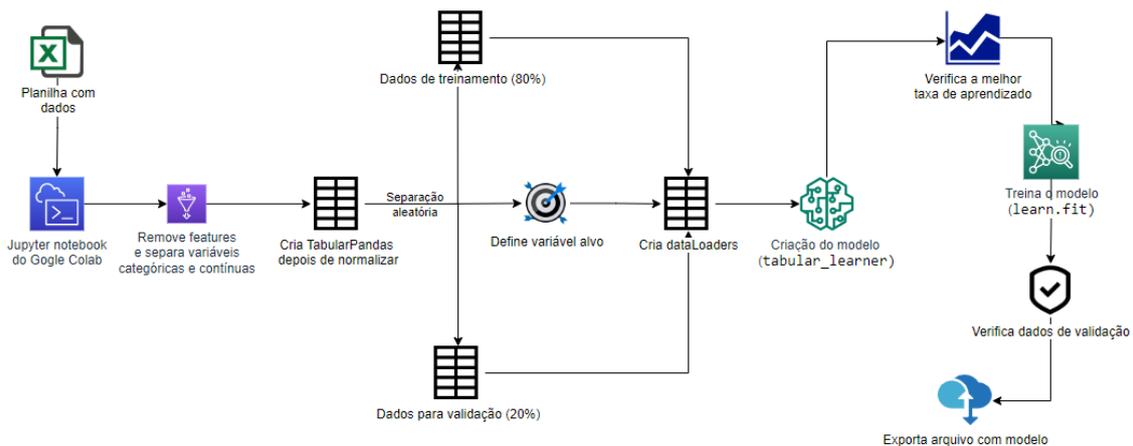


Figura 42 - Arquitetura do desenvolvimento do modelo de recomendação por classificação de dados tabulares

4.4. Implementação de modelo de recomendação

Após o modelo estar pronto é necessário exportá-lo para ser usado junto ao Sistema de Gestão de TCCs do INE. Como a utilização de código Python dentro de um sistema desenvolvido todo em Java não seria algo simples, foi optado por criar um micro serviço que faria a parte da recomendação, com o código em Python, tendo a com:

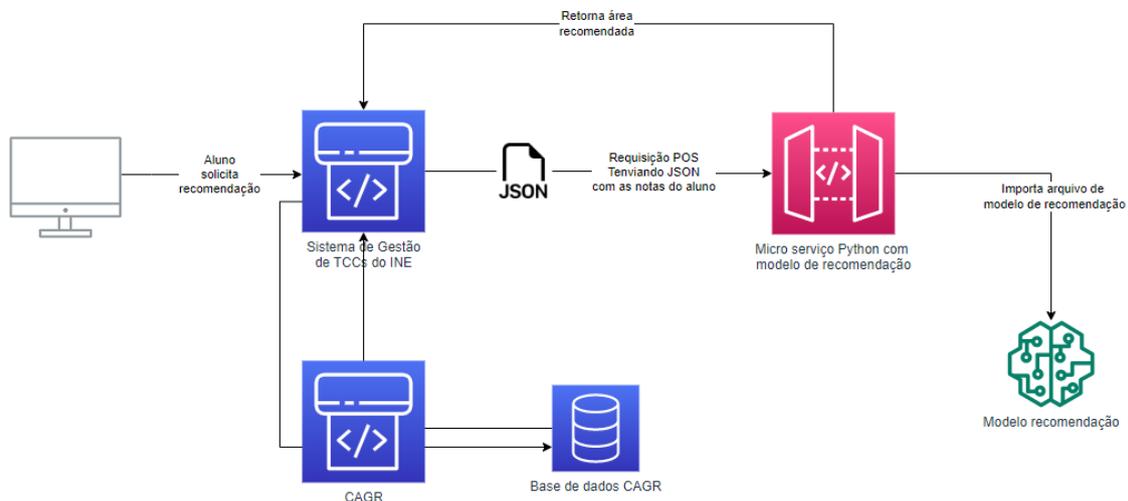


Figura 43 - Arquitetura do serviço de recomendação

Com o modelo treinado e extraído, tendo um micro serviço respondendo de maneira satisfatória, pode-se realizar a implementação de código para executar a recomendação e mostrar o resultado em tela. Junto com o desenvolvimento do módulo de recomendação, foram realizados testes da implementação de código, tanto de tela quanto de regra de negócios, dos cards de propostas de projetos e da associação entre um TCC, no momento que é cadastrado, com um tema de projeto proposto pelo professor.

Durante os testes foi necessário realizar a alteração de estrutura da atual base de dados, seja das tabelas já existentes ou das tabelas criadas, ao longo do desenvolvimento, bem como

incremento de lógica em diversos pontos já existentes do código. Também houve a necessidade de testar a comunicação entre o código fonte e o micro serviço de recomendação através de uma API, tendo que validar os dados de entrada e os dados de saída dessa chamada.

5. Avaliações

Nesta etapa, o objetivo é avaliar a corretude das recomendações de área de projeto para TCCs e a avaliação do módulo por parte dos usuários.

5.1. Avaliação do modelo

No código de criação do modelo de recomendação o *dataset* é dividido entre dados para treinamento e dados para teste. No treinamento do modelo foi realizado testes com diversos números de épocas, verificando que com 30 épocas os números ficaram o mais otimizado possível, para os diversos testes realizados, não fazendo sentido deixar o modelo com mais do que isso. Assim, com esse número de épocas, foi possível obter o valor bem próximo a 0,85 para a acurácia, o que indica um bom resultado, já que a acurácia está bem próxima de 1, que seria o valor ideal. Também se notou um valor próximo de 0,1 para o *train_loss*, que representa a diferença entre o valor de saída da rede e o esperado para uma determinada entrada. A diminuição gradativa do valor de *loss* a cada *epoch* indica que a rede está obtendo mais sucessos ao fazer previsões. Neste caso, quando mais próximo de 0, melhor.

Logo após foram feitos diversos testes manuais, com uma tabela com dados conhecidos que posteriormente, foram parcialmente apagados. A tabela foi utilizada como input para o sistema de recomendação. Para cada caso conhecido da planilha, foi gerada uma recomendação. Apesar de, nesse caso, a recomendação ter acertado todos os resultados, nem todos os testes isso ocorreu e não é garantido que o modelo sempre acertará um dado conhecido. A Figura 5 ilustra o último teste realizado com uma tabela que possuía vinte tuplas de dados conhecidos. Nela, pode-se observar que dos 20 dados, apenas 2, os destacados em vermelho, o modelo não conseguiu prever o valor correto.

AREA	AREA RECOMENDADA	ORIENTADOR	CLASSIF_AREA
Redes de Computadores e Segurança	Redes de Computadores e Segurança	Carlos Becker Westphall	INES429 - Segurança em Computação
Redes de Computadores e Segurança	Redes de Computadores e Segurança	Carlos Becker Westphall	INES429 - Segurança em Computação
Engenharia de Software, Gerência de Projetos	Engenharia de Software, Gerência de Projetos	Fabiane Barreto Vavassori Benitti	INES614 - Engenharia de Software
Teoria da Computação e Linguagens	Teoria da Computação e Linguagens		INES624 - Engenharia de Usabilidade
Engenharia de Software, Gerência de Projetos	Engenharia de Software, Gerência de Projetos	Christiane Anneliese Gresse Von Wangenheim	
Fundamentos de Programação	Fundamentos de Programação	Frank Augusto Siqueira	INES646 - Programação para Web
Tratamento e Banco de Dados	Engenharia de Software, Gerência de Projetos		
Tratamento e Banco de Dados	Tratamento e Banco de Dados		INES650 - Web Semântica, Ontologias e Sistemas de Informação
Fundamentos de Programação	Fundamentos de Programação		
Teoria da Computação e Linguagens	Engenharia de Software, Gerência de Projetos		
Tratamento e Banco de Dados	Tratamento e Banco de Dados		
Ciência, Tecnologia, Sociedade e Empreendedorismo	Ciência, Tecnologia, Sociedade e Empreendedorismo	Elder Rizzon Santos	INES428 - Informática e Sociedade
Tratamento e Banco de Dados	Tratamento e Banco de Dados	Ronaldo dos Santos Mello	INES454 - Tópicos Especiais em Gerência de Dados
Tratamento e Banco de Dados	Tratamento e Banco de Dados		
Engenharia de Software, Gerência de Projetos	Engenharia de Software, Gerência de Projetos	Patricia Vilain	INES455 - Testes de Software
Engenharia de Software, Gerência de Projetos	Engenharia de Software, Gerência de Projetos	Patricia Vilain	INES455 - Testes de Software
Teoria da Computação e Linguagens	Teoria da Computação e Linguagens	Elder Rizzon Santos	
Teoria da Computação e Linguagens	Teoria da Computação e Linguagens		INES633 - Sistemas Inteligentes
Engenharia de Software, Gerência de Projetos	Engenharia de Software, Gerência de Projetos	Jean Carlo Rossa Hauck	INES614 - Engenharia de Software

Figura 44 - Recorte de tabela usada para validação da recomendação

5.2. Avaliação do módulo de recomendação

A avaliação do módulo é realizada por usuários e é, de maneira simplificada, uma forma sistemática de observar o usuário usando uma ferramenta e assim, coletar informações específicas que permite determinar se a ferramenta foi posta de maneira fácil ou difícil para eles. Assim são realizados os testes de usabilidade para conseguir informações através dos dados gerados. A avaliação pode ser separada em duas visões, a do professor a do aluno. Para a avaliação do professor, foi realizada uma entrevista com o Coordenador de projetos de conclusão de curso e administrador do Sistema de Gestão de TCCs do INE, logo após a apresentação e uso do novo módulo. Com a entrevista, foram anotados pontos de melhorias e dúvidas pertinentes para eventuais correções ou melhorias do sistema. Já para os alunos, foi aplicado um questionário avaliando utilidade, desempenho, funcionalidade e a usabilidade, após o uso do módulo.

A execução da avaliação do módulo de recomendação foi realizada por alunos do INE cursando a disciplina de Introdução a Projetos, que precede a elaboração do TCC, momento no qual tipicamente os alunos estão selecionando um tema de TCC, e pelo professor coordenador de TCCs do INE. No total foram realizadas 11 avaliações, sendo uma com um professor que é a pessoa responsável pela administração do Sistema de Gestão de TCC do INE e as demais 10 avaliações, foram de alunos potenciais usuários do módulo. A avaliação realizada pelos alunos foi acompanhada pelo autor do trabalho, utilizando seu próprio computador de desenvolvimento para a execução do sistema Gestão de TCCs em ambiente de homologação.

Como resultado da pesquisa de satisfação temos que os usuários consideram que o módulo de recomendação é útil para os alunos e que, em sua maioria, concordam com a possibilidade de usar um tema de pesquisa indicado na recomendação. De forma geral, os alunos disseram que um direcionamento de área ou indicação de proposta de projeto ajuda muito e elogiaram a ideia e a proposta do trabalho. Em relação a funcionalidade, o sistema de recomendação teve um feedback positivo. Os alunos, em sua maioria, tiveram o entendimento de que a área recomendada e a informações disponíveis, resultante da recomendação, são suficientes para auxiliá-los. Quanto a usabilidade, pode-se observar um feedback positivo na facilidade de uso perante os alunos. Foi notado um tempo de observação dos *cards* apresentados antes de clicarem no botão para realização da recomendação personalizada, mas o que não caracteriza uma dificuldade na utilização. A média de notas atingida no emprego da abordagem SUS (BROOKE, 1996) para avaliação da usabilidade foi de 91,83, sendo uma nota bem alta. Já na questão de desempenho do módulo de recomendação não foi totalmente satisfatória devido a dificuldades enfrentadas com o ambiente de homologação do sistema. Por conta do *login* centralizado da UFSC, não foi permitido que os alunos acessassem o sistema, que estava em execução no computador de desenvolvimento, pelos computadores do laboratório onde as avaliações seriam realizadas, tendo que fazer todos os testes no computador de desenvolvimento e com o auxílio do autor do trabalho. Também não foi possível validar o módulo com os dados de histórico real dos alunos, pois nem todos estavam cadastrados no ambiente de homologação.

Área recomendada

Área Recomendada:
Engenharia de Software, Gerência de Projetos

Professores e temas:

Professores com interesse na área recomendada:
 Christiane Anneliese Gresse Von Wangenheim - Frank Augusto Siqueira - Joao Candido Lima Dovicchi - Mauricio Floriano Galimberti - Raul Sidnei Wazlawick - Ricardo Pereira e Silva - Fabiane Barreto Vavassori Benitti - Jean Carlo Rossa Hauck - Jose Eduardo de Lucca - Patricia Vilain -

Temas de pesquisa abertos com a área recomendada.



Área: Engenharia de Software, Gerência de Projetos (Macro)
Descrição: O trabalho consiste no desenvolvimento de um sistema web de recomendação de filmes que utiliza técnicas de filtragem colaborativa e filtragem baseada ...
Professor(a): Renato Cislighi
[Mais detalhes](#)



Área: Engenharia de Software, Gerência de Projetos (Macro)
Descrição: Será desenvolvido uma nova aplicação WEB para o suporte à Coordenação de Projetos com tecnologias atuais e uma interface mais agradável aos padrões at...
Professor(a): Jean Carlo Rossa Hauck
[Mais detalhes](#)



Área: Engenharia de Software, Gerência de Projetos (Macro)
Descrição: Desenvolver um aplicativo mobile com o intuito de agilizar o processo de registro de frequência dos alunos no ambiente virtual de ensino e aprendizagem...
Professor(a): Jean Carlo Rossa Hauck
[Mais detalhes](#)

Figura 45 - Tela de resultado da recomendação para determinado aluno

6. Conclusão

O presente trabalho apresenta o desenvolvimento de um módulo de recomendação de temas e projetos baseado em *Machine Learning* para o Sistema de Gestão de TCCs do INE. Para o desenvolvimento do trabalho foi inicialmente realizado o estudo da literatura, levantamento do estado da arte, a coleta e preparação do *dataset*, treinamento do modelo de IA e o desenvolvimento do módulo.

A implementação de um módulo de recomendação de temas de TCC para o sistema Gerenciamento de TCCs do INE passou por vários pontos da estrutura do sistema, tendo desde criação e alteração de tabelas da base de dados, criação e alteração de *view*, *controllers* e *models*, até a criação de um subsistema em Python que, ao receber os dados do aluno, executa o modelo de recomendação, com esses dados como input, e tem como resposta uma área que será passada para o sistema e exibida em tela, além de ser usada como filtro para listagem de professores e proposta de projetos.

Uma avaliação inicial do módulo desenvolvido, levantou indícios de que as informações resultantes da recomendação são satisfatórias e suficientes para auxiliar os alunos que estão, de certa forma, perdidos em relação a qual assunto estudar no seu trabalho de conclusão de curso. Também foi constatado que os *cards* de proposta de projetos exibidos, sejam eles filtrados pela área recomendada, ou não, vão ajudar tanto os professores a conseguirem alunos para seus projetos de pesquisa quanto os alunos, que estão em busca de alguma proposta. A avaliação também visou saber sobre a usabilidade, funcionalidade e desempenho do módulo e, de acordo com os dados obtidos nas respostas do questionário, verificou-se um resultado satisfatório por parte dos alunos.

Como trabalhos futuros, sugere-se, primeiramente, a implantação do novo módulo no ambiente de produção. Com isso, é importante que seja feita uma nova avaliação, agora em ambiente de produção. Também é recomendado que seja implementado um retreinamento automatizado do modelo de recomendação a cada semestre, ou seja, com o término de um semestre algum gatilho acionaria uma extração de dados estruturados da base de dados, passando pelo código de treinamento do modelo de recomendação e exportando o modelo para substituir o atual. Também pode-se pensar em aprimorar o sistema de recomendação usando além da classificação tabular, algum tipo de ontologia, ou até mesmo tendo mais fontes de dados para servir de input para o modelo, como os dados dos questionários de avaliação de disciplinas e professores que ocorre aos finais de cada semestre, ou até mesmo de algum questionário, dentro do próprio Sistema de Gestão de TCCs do INE, para avaliar a compatibilidade entre o aluno e professores orientadores de determinada área.

Referências

BOTELHO, Felipe Gonçalves; UGIONI, Pedro Henrique Rocha. **TCC UFSC - Nova Aplicação WEB para suporte à Coordenação de Projetos**. 2015. 67 f. TCC (Graduação) - Curso de Ciências da Computação, Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis, 2015.

FIOCRUZ, **Google Colab**, 2021. Disponível em: < <https://bigdata.icict.fiocruz.br/google-colab> >. Acesso: setembro 2021.

GONÇALVES, Diego Fretta. **Melhoria da usabilidade do sistema de tcc do ine/ufsc do ponto de vista do aluno**. 2016. 117 f. TCC (Graduação) - Curso de Ciências da Computação, Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis, Sc, 2016.

JUPYTER, **The Jupyter Notebook**, 2021. Disponível em: < <https://jupyter.org/> >. Acesso: setembro 2021.

MORANDINO, Matheus; RODRIGUES, Michelly. **Algoritmos de recomendação: mais presente no dia a dia do que você pensa**, 2020. Disponível em: <<http://www.each.usp.br/petsi/jornal/?p=2684>>. Acesso: setembro 2021.

ONGHERO, Eduardo Demeneck. **Implementação de Melhorias no Sistema de TCCs do Departamento de Informática e Estatística da UFSC: Perfil Professor**. 2018. Disponível em: < <https://repositorio.ufsc.br/handle/123456789/192165> >. Acesso: setembro 2021.

RUSSELL, STUART J. (STUART JONATHAN), 1962- **Inteligência artificial** / Stuart Russell, Peter Norvig; tradução Regina Célia Simille. – Rio de Janeiro: Elsevier, 2013.

SANTANA. Marlesson. **Deep Learning para Sistemas de Recomendação (Parte 1) — Introdução**. Disponível em: <<https://medium.com/data-hackers/deep-learning-para-sistemas-de-recomenda%C3%A7%C3%A3o-parte-1-introdu%C3%A7%C3%A3o-b19a896c471e>>. Acesso em: setembro 2021.

WAZLAWICK, Raul Sidnei. **Metodologia de Pesquisa Para Ciência da Computação**. Rio de Janeiro: Elsevier, 2009.