UNIVERSIDADE DE SÃO PAULO

Escola de Engenharia de São Carlos

Revelando a Technovation: uma exploração de dados da escola de verão para meninas por meio da ciência de dados

Marcelo Lopes Valerio

São Carlos – SP

Revelando a Technovation: uma exploração de dados da escola de verão para meninas por meio da ciência de dados

Marcelo Lopes Valerio

Orientadora: **Profª. Drª. Kalinka Regina Lucas Jaquie Castelo Branco**

Monografia final de conclusão de curso apresentada ao Curso de Engenharia de Materiais e Manufatura, da Escola de Engenharia de São Carlos – EESC-USP, como requisito parcial para obtenção do título de Engenheiro de Materiais e Manufatura.
Área de Concentração: Ciência de dados, Aprendizado de máquina, Raspagem de dados, Computação, Estatística.

USP – São Carlos
Maio de 2024

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da EESC/USP com os dados inseridos pelo(a) autor(a).

Valerio, Marcelo Lopes
V164r Revelando a Technovation: uma exploração de dados da escola de verão para meninas por meio da ciência de dados / Marcelo Lopes Valerio; orientadora Kalinka Regina Lucas Jaquie Castelo Branco. São Carlos, 2024.

Monografia (Graduação em Engenharia de Materiais e Manufatura) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2024.

1. Ciência de Dados. 2. Programas Educacionais. 3. Aprendizado de Máquina. 4. STEM. I. Título.

Eduardo Graziosi Silva - CRB - 8/8907

FOLHA DE APROVAÇÃO

Candidato / *Student*: Marcelo Lopes Valerio
Título do TCC / *Title*: Revelando a Technovation: uma exploração de dados da escola de verão para meninas por meio da ciência de dados
Data de defesa / *Date***:** 21/06/2024

Comissão Julgadora / *Examining committee*	**Resultado** / *Result*
Professora Kalinka Regina Lucas Jaquie Castelo Branco (orientador)	Aprovado
Instituição / *Affiliation*: ICMC **-** SSC	
Professora Lina Maria Garces Rodriguez	Aprovado
Instituição / *Affiliation*: ICMC **-** SSC	
Mestre Isadora Garcia Ferrão	Aprovado
Instituição / *Affiliation*: ICMC - SSC	

Presidente da Banca / *Chair of the Examining Committee*

Professora Kalinka Regina Lucas Jaquie Castelo Branco

AGRADECIMENTOS

Conforme chego ao fim da minha graduação, representado pela seguinte monografia, gostaria de deixar meus mais singelos agradecimentos às pessoas envolvidas ao longo desses quase 6 anos de trajetória.

Primeiramente, gostaria de agradecer à Escola de Engenharia de São Carlos e especificamente ao Departamento de Engenharia de Materiais e Manufatura pela oportunidade da minha vida de estudar em um ambiente tão saudável e inspirador, que valoriza a criatividade e principalmente a habilidade que estamos no mundo para exercer, independente da profissão: resolver problemas.

Gostaria de agradecer aos meus pais, Marcelo e Andrea, que nunca deixaram faltar nada para mim, tanto no quesito material, mas principalmente no quesito mental, me apoiando em todas as minhas empreitadas e ideias malucas que eu pudesse inventar. Vocês são a minha maior inspiração, o motivo de hoje em dia eu ser quem sou, e se, um dia conseguir ser metade do homem e da mulher que são, já estarei mais que feliz com minha vida. De longe meu maior exemplo tanto quando o assunto é dedicação, trabalho duro, e sentar a bunda e fazer o que precisa ser feito, quanto quando o assunto é amor e ajudar o próximo.

Aos meus irmão, Eduardo, Renan e Bruno, que desde minha primeira lembrança estão ao meu lado, seja para brigar, seja para brincar, conversar, ou fazer literalmente qualquer coisa. São meus melhores amigos, as pessoas com quem divido minha personalidade, e meu refúgio. Não iria tão longe se não fosse por eles, que sempre me apoiaram e me ajudaram, certamente meus melhores amigos, que espero ainda dividir muito da minha vida para que possamos compartilhar cada um a sua conquista, e todos nós comemorarmos.

Agradecimentos especiais à minha mentora e orientadora nesse projeto, Kalinka, que tive a sorte de conhecer tanto pessoal quanto profissionalmente, e é certamente uma fonte de inspiração pra mim. Demonstrou de maneira exemplar o que é o resultado de uma vida de esforço e trabalho duro, e conquistou, merecidamente, uma posição de respeito na instituição que dedica a vida, e ainda consegue equilibrar e impulsionar tudo isso com projetos pessoais de grande impacto na sociedade, como o objeto dessa monografia, o Technovation, entre muitos outros.

À organização do Technovation, por duas ocasiões; a primeira, por me permitirem ser mentor de um projeto tão impactante e maravilhoso que é essa iniciativa, não há nada nesse mundo que pague a sensação de poder ajudar o próximo, ainda mais fazendo isso da melhor

maneira possível: através da educação; e também por me permitirem estudar o projeto de perto, sua estrutura, analisar dados, e basear meu trabalho de conclusão, que representa tudo que eu aprendi durante meu tempo na faculdade, nesse projeto. Especialmente à Isadora Ferrão, parte da organização, que me auxiliou na obtenção de dados para o projeto, mas principalmente na criação dos gráficos e tabelas apresentados neste papel, sobre o programa.

Agradecer também às minhas avós, Dona Cida e Dona Catarina, as líderes de cada lado da família, que me ensinaram muito sobre amor ao próximo e resiliência, sempre me apoiando e me fazendo me sentir especial. Desde pequeno cuidando de mim, sempre estarão em meu coração.

Não poderia deixar de agradecer também aos meus colegas de classe, e amigos que fiz durante a faculdade, que tornaram as aulas mais leves, as noites de estudos menos desafiadoras, e os trabalhos mais interessantes, foi um prazer poder fazer parte vida de todos.

Por último, um agradecimento ao *International Office* da EESC, em conjunto com a Óbuda University, que me proporcionaram a chance única na vida de participar de um intercâmbio cultural e intelectual, onde eu pude consolidar as habilidades que hoje exerço como profissão e fazem parte deste trabalho final, a computação.

“Cada um de nós é, sob uma perspectiva cósmica, precioso.
Se um humano discorda de você, deixe-o viver.
Em cem bilhões de galáxias, você não vai achar outro como ele.”
(Carl Sagan)

RESUMO

VALERIO, M. L.. **Revelando a Technovation: uma exploração de dados da escola de verão para meninas por meio da ciência de dados**. 2024. 72 f. Monografia (Graduação) – Escola de Engenharia de São Carlos (EESC/USP), São Carlos – SP.

O projeto proposto tem como objetivo realizar uma análise de dados abrangente da *Technovation Summer School for Girls*, um programa de extensão, em formato de escola, voltado para o desenvolvimento de habilidades em ciência e tecnologia para garotas de 8 a 18 anos. Por meio da coleta e análise de dados relacionados ao programa, tem-se como objetivo a obtenção de informações valiosas sobre o impacto da *Techschool* e permitir predições que auxiliem em futuras edições, melhorando assim sua realização. Utilizando métodos de ciência de dados, são explorados diferentes conjuntos de dados, como registros de participantes, resultados de avaliações e registro de mentores. São aplicadas técnicas estatísticas e ferramentas de visualização de dados para identificar padrões, tendências e correlações significativas. Espera-se que os resultados dessa pesquisa forneçam informações valiosas para aprimorar futuras edições da *Techschool*, identificar áreas de melhoria para embasar decisões estratégicas para a expansão do programa, e apresentar dados para avaliar o impacto do programa desde a sua criação até o momento atual. Além disso, este estudo contribui para a área de educação em ciência, tecnologia, engenharia e matemática (STEM), fornecendo informações sobre a eficácia de programas no empoderamento e capacitação para jovens mulheres. Espera-se ainda que essa pesquisa contribua para o avanço do conhecimento sobre a importância de programas educacionais direcionados a meninas no campo da ciência e tecnologia, além de fornecer informações relevantes para aprimorar a experiência das participantes do *Technovation Summer School for Girls*. Por último, o trabalho apresenta o desenvolvimento de um modelo de triagem de mentores, construído com base nos dados coletados nos anos anteriores, o que visa reduzir o esforço na seleção dos candidatos mais adequados, desconsiderando os não adequados para o treinamento das meninas.

Palavras-chave: Ciência de Dados, Programas Educacionais, Aprendizado de Máquina, STEM.

ABSTRACT

VALERIO, M. L.. **Revelando a Technovation: uma exploração de dados da escola de verão para meninas por meio da ciência de dados**. 2024. 72 f. Monografia (Graduação) – Escola de Engenharia de São Carlos (EESC/USP), São Carlos – SP.

The proposed project aims to conduct a comprehensive data analysis of the Technovation Summer School for Girls, an outreach program focused on developing science and technology skills for girls aged 8 to 18. By collecting and analyzing data related to the program, the objective is to obtain valuable insights into the impact of Technovation on the participants' development. Using data science methods, different datasets such as participant records, assessment results, and mentors records will be explored. Statistical techniques and data visualization tools will be applied to identify patterns, trends, and significant correlations. The results of this research are expected to provide valuable insights for enhancing future editions of Technovation, identifying areas for improvement to support strategic decisions for program expansion, and presenting data to evaluate the program's impact since its inception. Additionally, this study may contribute to the field of science, technology, engineering, and mathematics (STEM) education by providing information on the effectiveness of programs in empowering and capacitating young women. It is also anticipated that this research will contribute to advancing knowledge about the importance of educational programs targeted at girls in the fields of science and technology, as well as providing relevant information to enhance the experience of participants in the Technovation Summer School for Girls. Lastly, the work will present the development of a mentor screening model, based on data collected in previous years, aiming to reduce effort in selecting the most suitable candidates, disregarding those considered unsuitable for training girls.

Key-words: Data Science, Educational Programs, Machine Learning.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE CÓDIGOS-FONTE

LISTA DE ABREVIATURAS E SIGLAS

Ada Boost Adaptative Boosting
API ..... *Application Programming Interface*
EESC .... Escola de Engenharia de São Carlos
GRACE .. **GR**upo de **A**lunas nas **C**iências **E**xatas
HTML ... *Hyper Text Markup Language*
HTML ... *Hyper Text Markup Language*
ICMC .... Instituto de Ciências Matemáticas e de Computação
PLN ..... *Processamento de Linguagem Natural*
STEM .... *Science, Technology, Engineering, and Math*
SVM ..... *Suport Vector Machine*
Techschool *Technovation Summer School for Girls*
USP ..... Universidade de São Paulo
XPATH ... *XML Path Language*

SUMÁRIO

Capítulo 1

INTRODUÇÃO

1.1 Motivação e Contextualização

A equidade de gênero nas áreas de *Science, Technology, Engineering, and Math* (STEM) é uma preocupação global, evidenciada pelas persistentes disparidades na representação e participação das mulheres nessas disciplinas. Iniciativas educacionais que visam aumentar a participação feminina nessas áreas são fundamentais para promover uma sociedade mais justa. O estímulo à participação é crucial, uma vez que a falta de modelos femininos nessas áreas pode desencorajar as jovens a seguir carreiras nessas disciplinas, perpetuando o desequilíbrio de gênero na ciência (STOET; GEARY, 2018).

Apesar dos avanços recentes, as mulheres continuam sub-representadas, especialmente em tecnologia e engenharia. Estudos evidenciam as diversas barreiras sociais, culturais e institucionais que as mulheres enfrentam ao longo de suas carreiras nessas áreas (DASGUPTA; STOUT, 2014), incluindo estereótipos de gênero, falta de modelos femininos de sucesso e discriminação no ambiente de trabalho. Esses obstáculos contribuem para a persistência da desigualdade de gênero, limitando não apenas as oportunidades das mulheres, mas também o processo de inovação e desenvolvimento (BEDDOES; PANTHER, 2018).

A *Technovation Summer School for Girls* (Techschool) é uma iniciativa que busca enfrentar esses desafios, oferecendo oportunidades para o desenvolvimento de habilidades em ciência e tecnologia para meninas de 8 a 18 anos. Este programa não apenas oferece conhecimentos práticos em áreas como programação e empreendedorismo, mas também visa promover a autoconfiança e o empoderamento das participantes. O *Technovation* de modo maias amplo é uma iniciativa global, como comprova (TORRES, 2015) que incentiva meninas a ingressarem na área da tecnologia e empreendedorismo desde jovens, encorajando-as a desenvolverem projetos onde são protagonistas, desde a concepção até a implementação técnica. A *Techschool*, por sua vez, segue o curriculum do *Technovation* sendo uma ação nacional sediada no Instituto de Ciências Matemáticas e de Computação (ICMC) da Universidade de São Paulo (USP).

Durante a *Techschool*, iniciativa no Brasil pelo grupo **GR**upo de **A**lunas nas **C**iências **E**xatas (GRACE) - ICMC/USP, as garotas selecionadas são orientadas por mentores multidisciplinares, provenientes de áreas como tecnologia, educação e *design*, a desenvolverem uma aplicação para dispositivos móveis, com o objetivo de solucionar problemas em suas comunidades, como *bullying*, abandono de animais e educação infantil, entre muitos outros. Atualmente, em sua

sexta edição, a *Techschool* teve seu inicio em 2018, como um evento presencial, na cidade de São Carlos, porém, desde 2021 é realizada de forma virtual, sendo capaz, desse modo, de atingir mais meninas ao redor do Brasil. As alunas participantes são categorizadas em três grupos diferentes, iniciantes, júniores e sêniores, de acordo com a faixa etária, o que influencia no nível de exigência na entrega, e também pode envolver mais etapas intermediárias envolvendo, por exemplo, o uso de inteligência artificial, criação de um plano de negócios estruturado, entre outros desafios.

1.2 Objetivos

A proposta deste trabalho é aplicar técnicas de agrupamento, filtragem, análise e interpretação de dados para oferecer uma visão ampla do impacto do programa até o momento, tanto na sociedade em geral quanto na jornada individual de todos os participantes, incluindo alunas e mentores. Por meio de *insights* embasados em dados estatísticos, almeja-se fornecer informações valiosas aos organizadores do programa, visando direcionar a *Techschool* para ampliar sua abrangência e aumentar sua eficácia. Isso implica aprimorar não apenas os processos de seleção das candidatas, mas também fortalecer a inclusão e a afirmação da presença nacional do programa.

Além disso, o estudo propõe o desenvolvimento de um modelo de aprendizado de máquina para triagem de candidatos a mentores, utilizando parâmetros como área de atuação, experiência profissional, formação e qualificações dos candidatos. Essa ferramenta visa não somente agilizar o processo de preparação do programa e reduzir a carga de trabalho da equipe envolvida, mas também garantindo uma melhor organização e eficiência no desenvolvimento da *Techschool*, bem como assegurando a capacidade e multidisciplinaridade dos mentores, e consequentemente a qualidade da mentoria oferecida às meninas.

1.3 Organização

No Capítulo 2, é conduzida uma revisão da terminologia básica utilizada no projeto, juntamente com os principais conceitos e metodologias empregados na elaboração desta monografia, com um enfoque particular em técnicas de coleta, agrupamento de dados e aprendizado de máquina. Em seguida, no Capítulo 3, são apresentados os resultados e a evolução da pesquisa ao longo do tempo, abrangendo a coleta e análise dos dados coletados e a aplicação dos conceitos discutidos anteriormente no contexto atual do estudo. Por fim, no Capítulo 4, são apresentadas as conclusões deste trabalho, correlacionando os objetivos estabelecidos, os resultados obtidos e as limitações da pesquisa atual, além de sugerir possíveis direções para trabalhos futuros na área.

Capítulo 2

REVISÃO BIBLIOGRÁFICA

2.1 Considerações iniciais

O objetivo desta seção é fornecer uma compreensão abrangente dos conceitos teóricos e metodologias fundamentais que sustentam a pesquisa, oferecendo uma visão detalhada sobre o processo de análise de dados e suas diversas etapas. Inicialmente é apresentada uma discussão sobre a importância da ciência de dados e seu impacto na sociedade contemporânea, explorando como essa disciplina revolucionou a forma como organizações e indivíduos tomam decisões informadas. Em seguida, são abordadas as etapas principais do processo de análise de dados, desde a coleta e pré-processamento até a modelagem e interpretação dos resultados, destacando métodos de validação e visualização que garantem a robustez e clareza das análises. Além disso, é enfatizada a aplicação prática da ciência de dados em diferentes setores, ilustrando seu papel crucial na inovação e resolução de problemas complexos. Por fim, são discutidas as considerações éticas e de privacidade essenciais para a condução responsável da análise de dados, garantindo que os *insights* derivados respeitem os direitos e a confidencialidade dos indivíduos.

2.2 Introdução à ciência de dados

2.2.1 Ciência de dados na sociedade atual

A Ciência de Dados surge como uma disciplina essencial no mundo atual, como evidenciado em áreas como medicina (RADENKOVIC; KEOGH; MARUTHAPPU, 2019), e em negócios (PROVOST; FAWCETT, 2013), utilizada para manipular e extrair *insights* significativos de conjuntos massivos de informações, muitas vezes de escalas incompreensíveis à capacidade humana. Sua importância transcende os limites de organizações comerciais e acadêmicas, influenciando significativamente a forma como decisões e pesquisas são realizadas atualmente.

A Ciência de Dados é uma ferramenta poderosa, que, por meio da análise de grandes volumes de dados, é capaz de revela padrões ocultos, oferecer *insights* valiosos e possibilita prever tendências futuras. Como aponta (BELLAZZI; ZUPAN, 2008), foi demonstrado como a análise de dados pode ser aplicada na área da saúde para prever a progressão de doenças crônicas, permitindo intervenções preventivas mais eficazes e personalizadas. Na otimização de processos, em setores como o comércio eletrônico, em uma pesquisa conduzida por Kohavi

et al. (2012), foi explorado o uso de técnicas de análise de dados para melhorar a eficiência de algoritmos de recomendação de produtos, resultando em um aumento significativo nas taxas de conversão e satisfação do cliente. Além disso, a Ciência de Dados impulsiona a inovação e a descoberta, capacitando profissionais a resolver problemas complexos, aliada de equipamentos computacionais poderosos e ferramentas estatísticas, de maneiras inesperadas e eficientes.

Contudo, é imprescindível uma análise minuciosa das implicações éticas e sociais inerentes à Ciência de Dados (FERGUSON, 2017). A coleta e análise de dados podem levantar questões cruciais relacionadas à privacidade, equidade e justiça, exigindo uma abordagem ética e responsável por parte dos profissionais envolvidos nesse campo. Por meio dessa discussão, não apenas é possível obter uma compreensão mais ampla da Ciência de Dados, mas também é essencial repensar seu papel na sociedade, considerando seus limites éticos, desafios técnicos e as limitações enfrentadas. Nesse contexto, vários governos e instituições tomam a frente na tentativa de estipular limites à invasão e manipulação de dados de usuários (FERGUSON, 2017).

2.2.2 Etapas necessárias para o processamento de dados

Os dados podem ser utilizados para diversos fins, como alimentar modelos de aprendizado de máquina, criar *dashboards* informativos, armazenar informações valiosas e embasar decisões estratégicas. Para que todas essas etapas ocorram de maneira eficaz, são seguidos alguns passos cruciais no processo de análise de dados, incluindo desde a coleta dos dados, seja por ferramentas como formulários, *webscraping*, ou passivamente na execução de uma aplicação, o preprocessamento, formatação e preparação dos mesmos, a validação e limpeza do conjunto de dados até a aplicação de algoritmos de análise e a interpretação dos resultados obtidos. Essa jornada dos dados, desde sua origem até sua transformação em *insights* acionáveis, constitui o cerne da Ciência de Dados e evidencia seu papel fundamental na era da informação e da tomada de decisões baseadas em evidências.

2.3 Coleta de dados

A coleta de dados desempenha um papel fundamental na análise de dados, pois influencia diretamente a qualidade e a quantidade das informações disponíveis para estudo e interpretação. Estudos científicos têm destacado a importância da coleta de dados de alta qualidade para garantir resultados confiáveis e significativos. A seleção cuidadosa das fontes de dados é essencial para garantir a validade e a confiabilidade dos resultados obtidos (HAIR *et al.*, 2009). Segundo Hair *et al.* (2009), a necessidade de uma abordagem criteriosa e variada na coleta de dados, considerando fatores como representatividade da amostra e precisão das medidas, respeitando critérios estatísticos para não criar uma amostra tendenciosa é apontada como uma das etapas mais significativas de todo o processo de análise.

Na categorização dos métodos de coleta de dados, é possível distinguir entre abordagens

passivas e ativas (MAHER *et al.*, 2019). Os métodos passivos envolvem a coleta de dados sem a intervenção direta do pesquisador, enquanto os métodos ativos requerem uma ação deliberada para obter as informações desejadas. Entre os métodos passivos de coleta de dados, incluem-se a utilização de formulários *online*, *cookies* de rastreamento em *websites* e registros de transações comerciais. Esses métodos são caracterizados pela coleta de dados que ocorre de forma natural ou automática, sem a necessidade de interação direta com os indivíduos ou fontes de informação.

Por outro lado, os métodos ativos de coleta de dados demandam uma ação proativa por parte do pesquisador para obter as informações desejadas. Um exemplo comum é o *webscraping*, que envolve a extração de dados de páginas da web por meio de *scripts* ou ferramentas automatizadas. Outras técnicas ativas incluem entrevistas, observação direta e experimentação controlada, nas quais o pesquisador interage diretamente com os participantes ou fontes de dados para obter as informações necessárias.

2.3.1 Coleta via formulários

A coleta de dados por meio de formulários é uma prática comum e amplamente utilizada em diversos contextos, incluindo pesquisas acadêmicas, estudos de mercado e, como mencionado, inscrições para eventos e programas. No contexto da *TechSchool*, todos os interessados, tanto alunas quanto mentores, precisaram preencher um formulário com dados pessoais, relevantes para a seleção. A escolha cuidadosa do que é perguntado no ato da inscrição é fundamental para a eficiência e a assertividade da seleção dos candidatos. Os formulários de inscrição oferecem uma maneira estruturada e eficiente de coletar informações importantes, como dados pessoais, histórico educacional, experiências anteriores e expectativas dos participantes em relação ao evento ou programa.

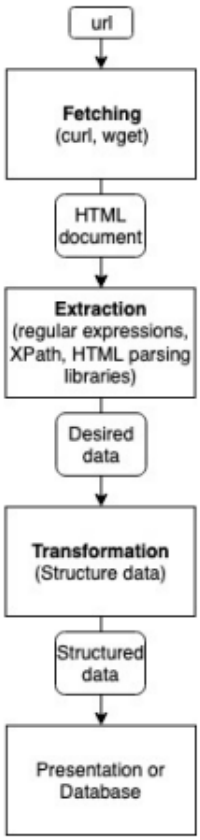
Esses formulários podem ser disponibilizados *online*, facilitando o acesso e aumentando o alcance do programa, sem limitá-lo a regiões físicas específicas, e a inscrição pode ser personalizada de acordo com as necessidades específicas de cada evento, permitindo a inclusão de perguntas adicionais para coletar informações específicas ou avaliar requisitos de elegibilidade, trazendo enormes benefícios, como citado em (BABBIE, 2020), nos tópicos sobre pesquisas *online*. A coleta de dados por meio de formulários de inscrição também oferece a vantagem de fornecer dados estruturados, que podem ser facilmente tabulados e analisados posteriormente, agilizando o processo posterior de normalização dos dados.

2.3.2 Webscraping

O *webscraping* é uma técnica amplamente discutida na literatura científica devido à sua relevância em várias áreas de pesquisa e aplicação, como utilizado em (SNELL; MENALDO, 2016), que aborda o *webscraping* no contexto da análise de *big data*, ressaltando sua importância para a obtenção de conjuntos de dados significativos e a aplicação de técnicas de análise de

dados avançadas. Também chamado, em português, raspagem de dados da web, é uma técnica utilizada para extrair informações de páginas da web de forma automatizada, amplamente empregada em diversas áreas, incluindo pesquisa acadêmica, análise de mercado, monitoramento de preços, entre outros. O processo de *webscraping* envolve o uso de programas de computador, *scripts* ou ferramentas automatizadas para percorrer páginas da web, identificar os dados de interesse e extrair essas informações de forma estruturada. Uma das vantagens do *webscraping* é a capacidade de coletar grandes volumes de dados de maneira rápida e eficiente. Isso permite aos pesquisadores e analistas acessar informações que de outra forma seriam difíceis ou demoradas de obter manualmente. Na Figura 1 é ilustrado o fluxograma de um código *webscraping*.

Figura 1 – Fluxograma básico de um código de *webscraping*



Fonte: (PERSSON, 2019)

Existem várias maneiras de extrair os dados, como utilizando linguagens de programação, além da criação do *script* do mesmo, na obtenção dos dados, necessitando conhecimento sobre a web e a estruturação do site alvejado, conhecendo linguagens como *Hyper Text Markup Language* (HTML) obtendo por meio do caminho até o elemento ou utilização de expressões regulares (PERSSON, 2019), para posterior registro do dado em alguma estrutura conhecida e que poderá ser facilmente trabalhada posteriormente.

No contexto da *Techschool*, o *webscraping* foi utilizado para coletar informações relevantes sobre os envolvidos no programa, uma vez providenciado o *LinkedIn* do candidato a mentor,

foi possível obter informações complementares aos formulários de inscrição, ajudando tanto na análise e seleção manual dos candidatos, evitando que os organizadores precisem procurar em várias páginas da internet, quanto possibilitando alimentar modelos e tabelas com dados já no momento da inscrição do mesmo.

No entanto, é importante ressaltar que o *webscraping* deve ser realizado com responsabilidade e ética. Algumas práticas de *webscraping* podem violar os termos de serviço de sites, infringir direitos autorais ou comprometer a privacidade dos usuários. Portanto, é fundamental garantir que o *webscraping* seja realizado de acordo com as leis e regulamentos aplicáveis, bem como com as políticas de uso aceitável dos sites de onde os dados são extraídos.

2.3.3 Coleta passiva durante a Techschool

A coleta passiva de dados ocorre sem a intervenção direta dos participantes ou pesquisadores, mas por meio da observação ou monitoramento de atividades realizadas pelos envolvidos. No contexto da *Techschool*, a coleta passiva de dados pode incluir registros automáticos de interações dos participantes com plataformas *online*, como registros de acesso a conteúdos educacionais, presença em reuniões e projetos da equipe, tempo de utilização de recursos digitais e padrões de engajamento com atividades propostas.

Por exemplo, ao utilizar uma plataforma *online* para disponibilizar materiais de estudo ou realizar atividades práticas, é possível coletar passivamente dados sobre quais recursos são mais acessados, quais tarefas são concluídas e quanto tempo os participantes dedicam a cada atividade. Essas informações podem fornecer *insights* valiosos sobre o envolvimento e o progresso das participantes ao longo do programa, ajudando os organizadores a avaliar a eficácia das estratégias de ensino e identificar áreas de melhoria.

Além disso, a coleta passiva de dados pode ser realizada por meio de ferramentas de análise de dados integradas às plataformas *online* utilizadas no programa, que registram automaticamente métricas relevantes, como taxas de conclusão de tarefas, taxas de retenção de participantes e padrões de engajamento ao longo do tempo. Esses dados podem ser utilizados para avaliar o desempenho do programa, identificar tendências de aprendizagem e personalizar a experiência dos participantes com base em seu comportamento *online*.

A coleta passiva de dados durante a *Techschool* não apenas fornece informações valiosas para os organizadores do programa, mas também pode melhorar a experiência dos participantes, garantindo que suas necessidades e interesses sejam atendidos de maneira mais eficaz e melhorando futuras edições.

2.4 Preprocessamento de Dados

O preprocessamento de dados é uma etapa essencial no fluxo de trabalho da Ciência de Dados, pois prepara os dados brutos para análise e modelagem. Este processo envolve uma série de procedimentos que limpam, transformam e organizam os dados, assegurando que estejam em um formato adequado para a aplicação de métodos estatísticos e algoritmos de aprendizado de máquina. A eficácia do preprocessamento é muito importante, uma vez que a qualidade dos dados influencia diretamente a precisão e a robustez dos modelos desenvolvidos.

Nesta seção, são exploradas as principais técnicas e metodologias de preprocessamento, incluindo limpeza de dados, tratamento de valores ausentes, normalização, padronização e transformação de dados. Além disso, são discutidos métodos para tratar os dados desbalanceados e a importância da seleção de características (*feature selection*) e da engenharia de características (*feature engineering*) para melhorar o desempenho dos modelos.

No contexto do algoritmo de *webscraping*, a seleção de características ajuda a criar um modelo eficiente e relevante, enquanto a engenharia de características transforma os dados de entrada em um formato mais adequado para o algoritmo.

A aplicação dessas técnicas no contexto da *Techshool* é detalhada, mostrando como essas práticas podem otimizar o processo de análise e fornecer percepções mais precisas e úteis para a avaliação e melhoria da escola. Isso inclui a implementação prática dessas metodologias para garantir que os dados coletados sejam de alta qualidade e bem preparados para as etapas subsequentes de análise e modelagem.

2.4.1 Limpeza e padronização de dados

A maior parte dos dados obtidos para as análises preliminares das alunas e mentores são resultados de formulários utilizados pela organização do evento, como formulários de inscrição, endereços, e controle de presença entre outros, de acordo com a disponibilidade ou abordagem utilizada pelos organizadores em cada uma das edições. O primeiro desafio foi extrair as informações de inscrição e participação dos eventos de maneira confiável, visto que as informações não seguem um padrão de ano para ano, podendo faltar campos relevantes entre as edições.

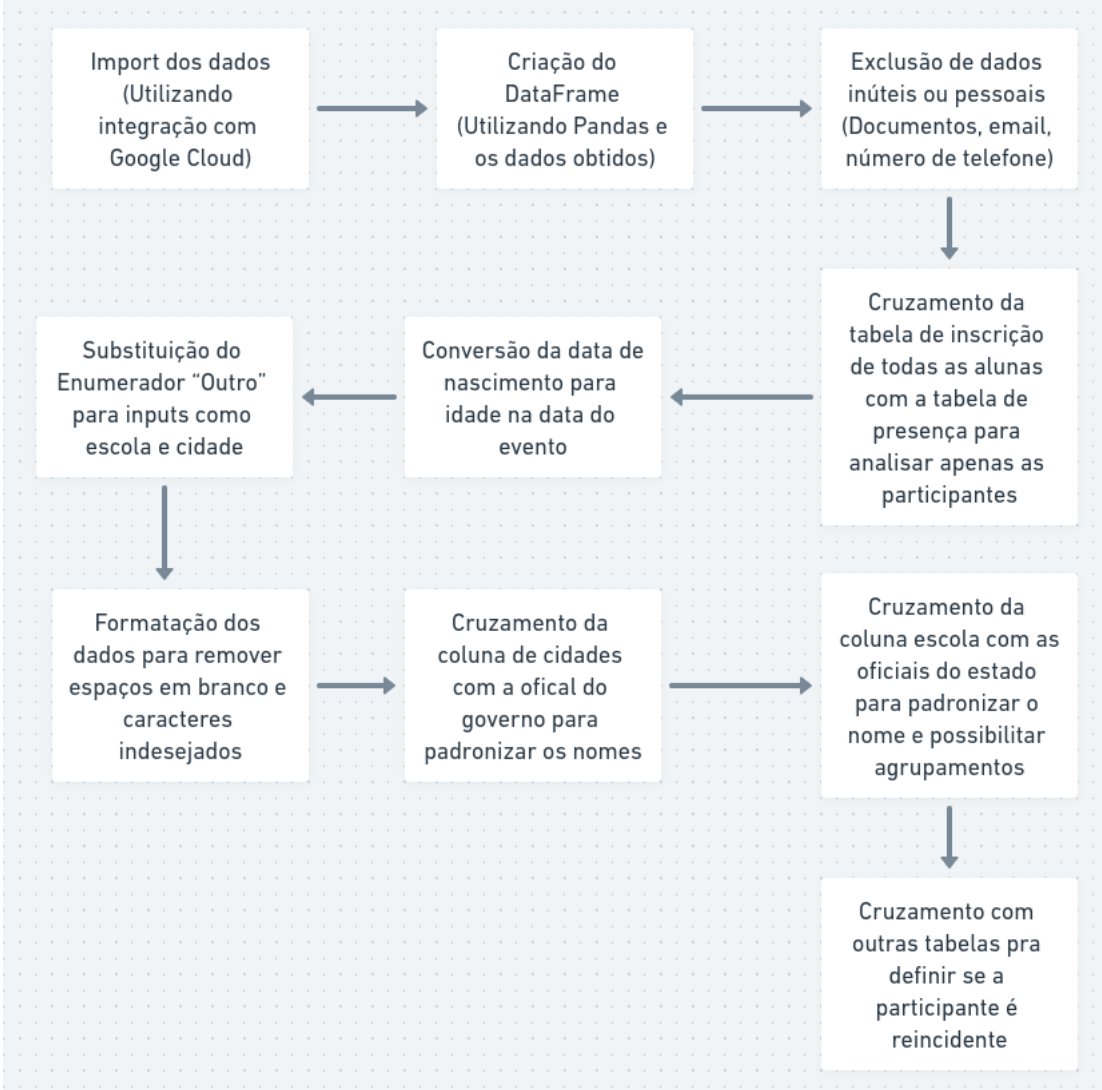
Para abordar essa questão, a extração dos dados foi realizada separadamente para cada edição do programa. Isso permitiu uma análise inicial de como os dados eram organizados e quais campos estavam disponíveis em cada ano. Foi montado um *DataFrame* em Pandas [1], uma biblioteca em python [2] utilizada para estruturar e manipular as informações de maneira simplificada e escalável.

[1] <https://pandas.pydata.org/docs/>
[2] <https://docs.python.org/3/>

A título de exemplo, é possível apontar que algumas edições incluíam detalhes sobre o histórico educacional dos participantes, enquanto em outras edições apenas o nome da escola. Essa variabilidade exigiu uma abordagem personalizada para cada conjunto de dados, identificando e padronizando as informações essenciais, para definir se a participante estuda em uma escola pública, ou privada, além da necessidade de deduzir a série escolar da mesma.

Na Figura 2 é apresentado o fluxograma do algoritmo utilizado na extração e formatação dos dados do *DataFrame* referente ao ano de 2020, que utilizou a planilha de inscrição, a planilha de presença, a planilha distribuída pelo governo com as cidades do estado de São Paulo, e a planilha distribuída pelo governo contendo nome e informações sobre as escolas do estado de São Paulo.

Figura 2 – Fluxograma do código utilizado na estruturação da tabela de participantes da edição de 2020.



Fonte: Elaborada pelo autor

Um ponto que vale ressaltar é a utilização da biblioteca *FuzzyWuzzy* [3], em um algoritmo

[3] <https://pypi.org/project/fuzzywuzzy/>

baseado no trabalho de Rao *et al.* (2018), que tem como objetivo minimizar impacto de dados mal formatados, ou similares, utilizada para agrupar os nomes das cidades e escolas de acordo com os dados fornecidos pelas candidatas. A biblioteca utiliza conceitos que são trabalhados a seguir, como *tokenização*, e cálculo de similaridade utilizando algoritmos matemáticos para então comparar o resultado com uma lista de valores e enquadrar a melhor combinação.

Para tratar os valores ausentes, foram utilizadas as planilhas externas de apêndice, como a de escolas, para definir a escola, que não estava categorizada, se era pública ou privada, por exemplo, preenchimento baseado em outras informações da aluna, como exemplificado, ao utilizar a idade da estudante para inferir o ano escolar que a mesma se encontra. Vale relembrar que cada edição teve seu próprio modo de processamento, visto que as planilhas utilizadas para coletar os dados não seguiam um padrão.

2.4.2 Seleção de Características e Engenharia de Características

Na modelagem e extração de dados, duas técnicas avançadas desempenham um papel importante: *Feature Selection* e *Feature Engineering*.

Feature Selection, ou seleção de características, refere-se à escolha das variáveis mais relevantes para a análise, o que pode ser feito por meio de algoritmos que ajudam a identificar quais colunas têm maior impacto nos resultados desejados. Essa técnica pode variar desde seleções simples até análises mais complexas, como a utilização de métodos estatísticos para avaliar a importância das variáveis ou a análise de correlação para identificar possíveis relações entre os dados que poderiam distorcer o modelo. Estudos mostram que uma seleção cuidadosa das características pode levar a modelos mais simples, eficientes e interpretáveis, sem comprometer a precisão dos resultados (GUYON; ELISSEEFF, 2003).

Já a Engenharia de Características (*feature engineering*) envolve a criação de novas variáveis ou transformações das características existentes para melhorar o desempenho dos modelos. Esta técnica é frequentemente aplicada para lidar com dados não-lineares ou para capturar melhor os padrões subjacentes nos dados. Um exemplo comum de engenharia de características é o método "get_dummies", amplamente utilizado em conjuntos de dados categóricos. Ele transforma uma coluna de dados categóricos em várias colunas binárias (0 ou 1), permitindo que algoritmos de aprendizado de máquina interpretem essas características de forma mais eficaz. Outras técnicas de engenharia de características incluem normalização, padronização, discretização e criação de recursos polinomiais. Pesquisas indicam que a engenharia de características bem projetada pode resultar em modelos mais robustos e generalizáveis, melhorando significativamente o desempenho preditivo (ZEBARI *et al.*, 2020).

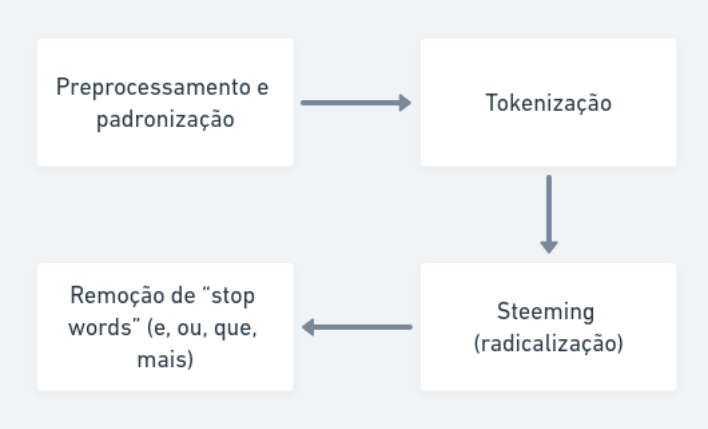
2.4.3 Tokenização e Steming

Tokenização e o *stemming* são técnicas fundamentais no pré-processamento de texto, amplamente utilizadas para análise de linguagem natural e mineração de texto (HEINES *et al.*, 2021). A tokenização envolve a divisão de um texto em unidades menores, chamadas *tokens*, que podem ser palavras individuais, sílabas ou caracteres. Essa técnica é essencial para preparar o texto para análises posteriores, como contagem de palavras, identificação de n-gramas e análise de frequência.

O *stemming*, por sua vez, refere-se à redução das palavras à sua forma base ou raiz, removendo sufixos e prefixos para extrair o significado básico da palavra (HEINES *et al.*, 2021). Isso ajuda a agrupar palavras relacionadas e simplificar a análise de texto, especialmente em casos onde diferentes formas da mesma palavra podem ser encontradas. Por exemplo, palavras como jogador e jogando podem ser radicalizadas à palavra base jogar.

A Figura 3 ilustra as etapas de preparação para alimentar um modelo de processamento de língua natural.

Figura 3 – Etapas de preparação para alimentação de um modelo de processamento de linguagem natural.



Fonte: Adaptada de (VIJAYARANI; JANANI *et al.*, 2016)

Essas técnicas são amplamente aplicadas em várias áreas, como processamento de linguagem natural, recuperação de informação e classificação de texto. Em estudos de caso, a tokenização e o *stemming* têm sido utilizados com sucesso para análise de sentimentos em mídias sociais, categorização de documentos e extração de informações de grandes conjuntos de dados textuais (SCHÜTZE; MANNING; RAGHAVAN, 2008). Além disso, essas técnicas são essenciais para o desenvolvimento de sistemas de busca eficazes, que dependem da compreensão precisa e eficiente do texto para retornar resultados relevantes aos usuários.

No projeto aqui desenvolvido, foram utilizados em conjunto ao algoritmo de *webscrapping* com o objetivo de categorizar os candidatos a mentores de acordo com a descrição profissional fornecida em seu perfil, tokenizando a descrição profissional ou a informação acadêmica do mesmo, alimentando um algoritmo de processamento de linguagem natural e

aprendizado de máquina, para encaixar o candidato em categorias consideradas relevantes para o desenvolvimento do modelo de seleção de mentores.

2.5 Aprendizado de Máquina

O aprendizado de máquina aparece como um subcampo da inteligência artificial, principalmente a partir da segunda metade do Século XX, onde foi possível observar a evolução e o desenvolvimento de algoritmos de auto-aprendizagem para adquirir conhecimento a partir desses dados, principalmente com o objetivo de fazer previsões. Pode-se dizer que o aprendizado de máquina estuda o desenvolvimento de métodos computacionais capazes de extrair conceitos, conhecimentos, habilidades e meios de organizar o conhecimento existente nas amostras de dados Raschka (2015), Murphy (2012). Em vez de exigir seres humanos para derivar regras manualmente e construir modelos de análise de grandes quantidades de dados, o aprendizado de máquina oferece uma alternativa mais eficiente para capturar o conhecimento a partir dos dados e melhorar gradualmente o desempenho de modelos preditivos, além de tomar decisões baseadas nesses dados. Sendo assim, o Aprendizado de Máquina tem se tornado cada vez mais importante na pesquisa científica, mas também tem desempenhado um papel relevante na vida cotidiana. De modo geral, os diferentes algoritmos de aprendizado de máquina são utilizados de forma a gerar classificadores para um conjunto de exemplos. Entende-se por um processo de atribuir a uma determinada informação o rótulo da classe a qual ela pertence (RUSSELL; NORVIG, 2002).

As técnicas de aprendizado de máquina são empregadas por indução, ou seja, a partir de um conjunto de treinamento, de um classificador, que por sua vez deve ser capaz de prever a classe de instâncias do domínio em que foi treinado.

A estrutura básica de um sistema de aprendizado de máquina é formada por quatro campos:

- ***Environment*:** Ambiente do sistema. É a parte que proporciona informações para a parte de aprendizado do sistema;
- ***Learning*:** Aprendizado. Responsável por revisar a base de conhecimento fazendo uso de informações do ambiente;
- ***Knowledge Base*:** Base de conhecimento. Vetor de características, sentenças lógicas, regras de modelo de produção, redes semânticas, entre outros.
- ***Execution*:** Constitui o núcleo de todo o sistema, sendo a parte operativa, cujo foco está no aperfeiçoamento das ações de aprendizagem.

Existem três tipos diferentes de técnicas de aprendizado de máquina, ou paradigmas, que podem ser utilizados na geração de um preditor: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço (RASCHKA, 2015).

A escolha do paradigma de aprendizado determina como o algoritmo de aprendizado de máquina se relaciona com seu meio ambiente e como ocorrerá o aprendizado.

Os paradigmas são compreendidos da seguinte forma:

- **Supervisionado**: Nesse paradigma, existe um "professor externo"que representa o conhecimento do ambiente por meio de conjuntos de exemplos na forma de entrada-saída. O algoritmo de aprendizado de máquina é treinado com exemplos rotulados da função a ser aprendida, permitindo que ele aprenda a representar ou agrupar as entradas submetidas com base em uma medida de qualidade.

- **Não supervisionado**: Aqui, não há presença de um "professor". Não existem instâncias rotuladas da função a ser aprendida. O algoritmo aprende a agrupar as entradas com base em uma medida de qualidade, sem orientação externa.

- **Por Reforço**: O aprendizado ocorre por meio de recompensas ou penalidades dadas ao algoritmo, dependendo do seu desempenho em aproximar a função desejada.

2.5.1 Aprendizado de máquina supervisionado

Os modelos supervisionados desempenham um papel fundamental no campo da aprendizagem de máquina, sendo amplamente utilizados em uma variedade de aplicações que vão desde previsão de vendas até diagnósticos médicos. Esses modelos são treinados com um conjunto de dados rotulado, onde cada instância de dados possui uma variável de saída associada, também conhecida como rótulo ou classe. Durante o treinamento, o modelo aprende a relação entre as variáveis de entrada e os rótulos correspondentes, permitindo fazer previsões ou classificações em novos conjuntos de dados (HASTIE *et al.*, 2009).

Uma das principais tarefas realizadas por modelos supervisionados é a regressão, onde o objetivo é prever um valor contínuo com base em um conjunto de variáveis de entrada. Por exemplo, um modelo de regressão pode ser treinado com dados históricos de vendas de uma empresa para prever as vendas futuras com base em fatores como publicidade, temporada e preço. Algoritmos comuns usados em regressão incluem regressão linear, regressão logística, árvores de decisão e *Suport Vector Machine* (SVM).

Outra tarefa importante é a classificação, onde o objetivo é atribuir uma classe ou categoria a uma instância de dados com base em suas características. Por exemplo, um modelo de classificação pode ser treinado com dados de pacientes para prever se um tumor é maligno ou benigno com base em características como tamanho, forma e textura. Algoritmos populares de classificação incluem árvores de decisão, k-vizinhos mais próximos (KNN), redes neurais e algoritmos de *ensemble* como o *Random Forest* e o Adaptative Boosting (Ada Boost), que combinam vários classificadores fracos (pouco desenvolvidos) para gerar um forte.

Uma das vantagens dos modelos supervisionados é sua capacidade de aprender a partir de dados rotulados, o que os torna adequados para uma ampla gama de problemas do mundo real. No entanto, é importante ressaltar que a qualidade dos resultados depende da qualidade dos dados de treinamento e do ajuste adequado dos parâmetros do modelo. Além disso, a interpretabilidade dos resultados e a capacidade de generalização para novos conjuntos de dados também são considerações importantes ao usar modelos supervisionados em aplicações práticas (BISHOP, 2006).

O aprendizado supervisionado foi o utilizado no desenvolvimento deste trabalho.

No Python existe uma gama de bibliotecas com modelos pré-construídos, prontos para serem utilizados, fornecidos pela biblioteca Python Scikit-Learn, muito utilizada para desenvolvimento de redes neurais e modelos de classificação e regressão (SCIKIT-LEARN. . . , 2024), como:

- *Logistic Regression*;
- *Decision Tree Classifier*;
- *AdaBoost Classifier*;
- *K-Neighbors Classifier*;
- *Random Forest Classifier*.

2.5.1.1 Avaliação de modelos de aprendizado de máquina

A *cross validation* é uma técnica essencial na avaliação de modelos de aprendizado de máquina, especialmente em situações onde se tem um conjunto de dados limitado e é necessário evitar o viés na seleção dos conjuntos de treinamento e teste. Em sua forma mais comum, a *cross validation k-fold* divide o conjunto de dados em k subconjuntos (ou *folds*) aproximadamente iguais. O modelo é treinado k vezes, cada vez usando k-1 *folds* como conjunto de treinamento e o *fold* restante como conjunto de teste. Isso resulta em k conjuntos de métricas de desempenho, que podem ser médias para fornecer uma estimativa geral do desempenho do modelo (HASTIE *et al.*, 2009).

Alinhada à *cross validation*, pode-se realizar análises estatísticas para a avaliação do desempenho do modelo sobre o *DataSet* de teste. Os principais parâmetros são:

- *A **acurácia** é a proporção de previsões corretas feitas pelo modelo em relação ao total de previsões feitas. É uma medida simples e intuitiva do desempenho do modelo (BISHOP, 2006).*

- *A **precisão** mede a proporção de previsões positivas corretas em relação ao total de previsões positivas feitas pelo modelo. É útil quando o foco está em minimizar os falsos positivos (SCHÜTZE; MANNING; RAGHAVAN, 2008).*

- *A **revocação** mede a proporção de instâncias positivas que foram corretamente identificadas pelo modelo em relação ao total de instâncias positivas presentes nos dados. É útil quando o foco está em minimizar os falsos negativos (SCHÜTZE; MANNING; RAGHAVAN, 2008).*

- *O **F1 Score** é a média harmônica da precisão e da revocação. Ele fornece uma medida única que leva em consideração tanto falsos positivos quanto falsos negativos. É uma métrica útil quando há um desequilíbrio entre as classes (MURPHY, 2012).*

2.5.1.2 Seleção de modelos de aprendizado de máquina

Para a seleção do modelo a ser utilizado, foi aplicada uma técnica chamada **grid search**, que consiste em realizar várias iterações de treino e teste, testando todas as combinações,com o objetivo de definir a melhor combinação de hiperparâmetros, que são basicamente parâmetros que são passados ao modelo para definir a estrutura do mesmo, simplificando uma interface complexa em alguns parâmetros ao instanciar o modelo (PROBST; BOULESTEIX; BISCHL, 2019), de acordo com um *score*, no caso, foi escolhida a métrica de **F1-score** para avaliar os modelos (LIASHCHYNSKYI; LIASHCHYNSKYI, 2019).

Foi criada então uma função para aplicar o algoritmo de *Grid Search* em todos os modelos apontados anteriormente, para então pontuá-los e selecionar o melhor modelo para classificar os mentores, como ilustrado na Figura 4.

Note que a função recebe como entrada a referência do modelo a ser testado, e os parâmetros a serem testados nessa tentativa. Após aplicar para todos os modelos, o escolhido para aprofundamento foi o *Decision Tree Classifier*, devido à sua capacidade de se adaptar aos dados, e desempenho satisfatório, como abordado na seção de resultados.

2.6 Visualização e análise estatística de dados

A visualização de dados é uma etapa crucial na análise de dados, especialmente quando se trata de explorar, interpretar e comunicar informações complexas. Por meio de gráficos e representações visuais, é possível detectar padrões, identificar tendências e entender a distribuição dos dados de forma mais intuitiva. No contexto do projeto da *Techschool* foram utilizadas as bibliotecas *Matplotlib* e *Seaborn* para criar visualizações eficazes e informativas.

Matplotlib é uma biblioteca poderosa e flexível para criar gráficos em Python, proporcionando uma ampla variedade de opções para personalização e formatação de gráficos (HUNTER, 2007). *Seaborn*, construída sobre o *Matplotlib*, oferece uma interface de alto nível que facilita

Figura 4 – Algoritmo utilizado para aplicar a técnica de *Grid Search* nos modelos analisados.

```
def evaluate_model(model_ref, param_grid):
  model = model_ref()
  custom_scorer = make_scorer(f1_score, pos_label=1)

  grid_search = GridSearchCV(estimator=model, param_grid=param_grid, cv=3, scoring=custom_scorer, verbose=1)

  sample_weights = np.ones_like(y_train)
  sample_weights[y_train == 1] = 5

  grid_search.fit(X_train, y_train, sample_weight=sample_weights)

  print("Best parameters:", grid_search.best_params_)
  print("Best score:", grid_search.best_score_)

  best_model = grid_search.best_estimator_
  predictions = best_model.predict(X_test)

  print(classification_report(y_test, predictions))

  conf_matrix = confusion_matrix(y_test, predictions)
  plt.figure(figsize=(4, 3))
  sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues", xticklabels=['0', '1'], yticklabels=['0', '1'])
  plt.show()
```

Fonte: Elaborada pelo autor

a criação de gráficos estatísticos atraentes e informativos (WASKOM, 2021). Utilizando essas duas bibliotecas, foram gerados gráficos que desempenharam um papel fundamental na análise e interpretação dos dados coletados.

Um dos gráficos mais impactantes utilizados foi o mapa de calor (*heatmap*), que permite a visualização de correlações entre diferentes variáveis. Essa técnica ajuda a identificar relações fortes ou fracas entre variáveis, o que é essencial para entender como diferentes fatores podem influenciar os resultados do modelo. Por exemplo, ao analisar a correlação entre o tempo de experiência no mercado de trabalho dos mentores e a categorização de aptos ou não para o programa, o mapa de calor revelou percepções valiosas que orientaram a não escolha desse parâmetro como relevante para o modelo, devido à sua fraca correlação.

Outra ferramenta visual importante foi a **matriz de confusão**, utilizada para avaliar o desempenho dos modelos de classificação (VISA *et al.*, 2011). A matriz de confusão fornece uma representação clara das previsões corretas e incorretas feitas pelo modelo, permitindo uma análise detalhada de sua precisão e áreas de melhoria. Essa visualização foi essencial para ajustar e melhorar os algoritmos de aprendizado de máquina, garantindo que os modelos fossem capazes de classificar os dados de forma mais eficaz, e possibilitando um comparativo com vários *subsets* do *dataset* original até encontrar o com melhor desempenho. Um exemplo de matriz de confusão é ilustrado na Figura 5.

Além disso, gráficos de dispersão, gráficos no formato "pie"e histogramas foram utilizados para visualizar a distribuição dos dados e suas características estatísticas. Gráficos de dispersão, por exemplo, ajudaram a identificar *outliers* e a compreender a relação entre variáveis

Figura 5 – Exemplo de output de uma matriz de confusão sobre o resultado de um modelo preditivo. Note que a mesma oferece medidas estatísticas e análises visuais e simplificadas, resultantes de análises profundas, do resultado do modelo.

	precision	recall	f1-score	support
0	0.81	0.38	0.52	92
1	0.24	0.69	0.36	26
accuracy			0.45	118
macro avg	0.53	0.54	0.44	118
weighted avg	0.69	0.45	0.48	118

Fonte: Elaborada pelo autor

contínuas. gráficos *pie* para a caracterização do *dataset* para variáveis com poucas possibilidades de valores, enquanto histogramas permitiram uma análise detalhada da distribuição de frequências.

A plotagem de medidas estatísticas, como média, mediana e desvio padrão, diretamente nos gráficos, acrescentou uma camada adicional de informação, facilitando a compreensão dos dados e destacando variações significativas. Essas visualizações ajudaram a detectar discrepâncias e a garantir que os dados fossem tratados de maneira adequada antes de serem utilizados na modelagem.

No geral, a utilização combinada de diferentes tipos de gráficos e visualizações foi decisiva para a construção de modelos robustos e precisos. As percepções obtidas por meio dessas técnicas orientaram não apenas a fase de pré-processamento e limpeza dos dados, mas também a seleção e ajuste dos modelos, contribuindo para a criação de algoritmos mais eficientes e eficazes.

2.6.1 Processamento de linguagem natural

O *Processamento de Linguagem Natural* (PLN) é um campo da inteligência artificial que se concentra na interação entre computadores e linguagem humana. Ele envolve o desenvolvimento de algoritmos e técnicas para permitir que computadores compreendam, interpretem e gerem linguagem humana de forma eficaz (BIRD; KLEIN; LOPER, 2009). O PLN abrange uma ampla gama de tarefas, desde a análise de sentimentos em textos até a tradução automática de idiomas e a geração de respostas automáticas em chatbots.

Essa disciplina multidisciplinar combina conceitos e técnicas de linguística, ciência da computação, estatística e inteligência artificial para desenvolver sistemas capazes de processar e entender a linguagem natural em suas diversas formas, como texto escrito, fala e gestos. O PLN tem aplicações em várias áreas, incluindo sistemas de busca, assistentes virtuais, análise de sentimentos em mídias sociais, tradução automática, sumarização automática de textos, entre outros.

As técnicas de PLN têm avançado rapidamente nos últimos anos, impulsionadas pelo

aumento da disponibilidade de dados, avanços em algoritmos de aprendizado de máquina e o desenvolvimento de modelos de linguagem cada vez mais sofisticados, como os modelos de linguagem baseados em redes neurais. No atual trabalho, foi utilizado para categorizar os candidatos de acordo com a área profissional de atuação, e também sua área acadêmica.

2.6.2 Considerações Finais

Neste capítulo foram apresentados os principais conceitos a serem utilizados neste trabalho. Foram discutidas a coleta de dados, o processamento de dados, bem como os conceitos de aprendizado de máquina e os tipos de algoritmos utilizados, seja por meio de bibliotecas prontas ou de códigos próprios. A visualização e a forma de realizar a análise estatística também foram detalhadas, proporcionando uma base sólida para o entendimento dos métodos e técnicas aplicadas no decorrer da pesquisa. Além disso, abordamos as ferramentas e métodos utilizados para implementar, testar e qualificar os modelos, destacando suas funcionalidades e importância no contexto do estudo.

O próximo capítulo tem como objetivo a apresentação do desenvolvimento do trabalho, onde serão descritas as etapas práticas realizadas, desde a implementação dos algoritmos até a execução dos testes e validações. Serão detalhados os resultados obtidos, incluindo gráficos que descrevam visualmente os dados apresentados. A discussão crítica dos achados permitirá avaliar a eficácia dos métodos utilizados e identificar possíveis melhorias ou futuras direções para pesquisas adicionais no contexto do programa da *Techschool*. Esta abordagem garantirá uma compreensão completa do processo e das conclusões derivadas deste estudo.

Capítulo 3

DESENVOLVIMENTO

3.1 Considerações iniciais

A seção de Desenvolvimento é necessária para compreender como a pesquisa foi conduzida e como os objetivos foram alcançados. Ela é estruturada em torno de dois tópicos principais: a criação de gráficos e análises sobre os mentores e alunas que já participaram da *Techschool*, fornecendo gráficos e dados para auxiliar a organização em edições futuras, e o desenvolvimento do modelo preditivo para a seleção de novos mentores. Cada um desses tópicos é abordado de maneira detalhada para demonstrar as metodologias, técnicas e ferramentas utilizadas ao longo deste trabalho.

No primeiro tópico, são apresentados os métodos de coleta e pré-processamento dos dados, seguidos por uma análise descritiva e estatística que inclui a distribuição das participantes por diferentes critérios, como localização geográfica, idade, e tipo de rede de ensino. A visualização de dados, utilizando bibliotecas como *Matplotlib* e *Seaborn*, destacadas pela importância em tornar os resultados mais compreensíveis e acionáveis.

No segundo tópico, o foco está no desenvolvimento do modelo preditivo, começando com a engenharia e seleção de características, passando pela avaliação de diversos modelos de aprendizado supervisionado, como *K-Nearest Neighbors*, *Random Forest*, *SVM* e *AdaBoost*, e culminando na seleção e implementação do modelo mais eficaz. São discutidas também as técnicas de avaliação de desempenho e a utilização de *Grid Search* para otimização dos modelos.

O objetivo desta seção é fornecer uma visão clara e detalhada das etapas de desenvolvimento da pesquisa, demonstrando como cada técnica e ferramenta contribuiu para atingir os objetivos propostos e proporcionando uma base sólida para a replicação e aplicação prática dos resultados obtidos.

3.2 Análise dos participantes

Como citado anteriormente, a principal fonte de dados utilizada neste trabalho foram as planilhas utilizadas pela organização do evento em edições passadas, por dois motivos: contar com dados confiáveis e pulverizados das participantes e mentores, e por conter dados atemporais, pois contam com o *status* durante a seleção na época do evento e ano em questão. Nessa seção, a

abordagem das meninas e dos mentores são demonstradas separadamente, por conter cada uma sua peculiaridade.

3.2.1 Análise técnica das alunas

3.2.1.1 Coleta de dados e montagem do dataset

Para alcançar um resultado final limpo e com dados valiosos, foi necessário cruzar várias planilhas, utilizar funções de processamento e padronização, e combinar os resultados individuais de cada ano para obter um conjunto de dados geral, representando todas as meninas influenciadas pela *Techschool*. Vale ressaltar que cada ano contou com processos ligeiramente diferentes, devido às peculiaridades e nuances do método de seleção. Para descrever a entrada dos dados, as colunas foram separadas em três categorias: colunas irrelevantes, que foram descartadas; colunas relevantes, utilizadas com pouco ou nenhum processamento; e colunas com dados "crus", que exigiram etapas intermediárias de transformação para se tornarem úteis para a pesquisa. A Tabela 1, segue uma relação das colunas que normalmente aparecem nas tabelas.

Tabela 1 – Relação de colunas por nível de utilidade nas planilhas da organização do programa. Note que as colunas não são constantes entre as planilhas, podendo ter mais ou menos colunas a depender da edição.

Colunas irrelevantes	**Colunas Relevantes**	**Colunas incompletas**
e-mail	Nome	Data de nascimento
Número de telefone	Categoria de inscrição	Nome da cidade
RG/ CPF	Já participou	Nome da escola
Dados/documentos do responsável	Estado	Período escolar
Termo de compromisso	Tipo de escola que estuda	-
Observações	-	-

Fonte: O autor.

O objetivo final é obter um conjunto de dados homogêneo que represente todas as garotas que foram alunas do programa, para então avaliar a abrangência e efetividade do *Techschool*, além de informações úteis como recorrência de alunas, quantidades de alunas de escola pública, entre outros abordados mais à frente nessa seção.

Como já apresentado, para alcançar esse objetivo final, foram criados vários *scripts*, tanto para formatar dados, aumentar a confiabilidade de entradas livres (caixas de texto, como o nome da cidade e escola, que dificultam agrupar por categoria, por exemplo), juntar com outras tabelas importantes e filtrar por dados relevantes, como apenas as alunas pertencentes à categoria Júnior.

Uma etapa que vale destacar foi a de padronização e categorização dos nomes das escolas. Por exemplo, uma mesma escola, nomeada por alunas da cidade de São Carlos, chamada Escola Estadual Conde do Pinhal, apresentou, pelo menos, cinco variações além da apresentada: E. E. Conde do Pinhal, EEEI Conde do Pinhal, E.E.E.I Conde do Pinhal, EE Conde do Pinhal, e o nome completo, Escola Estadual de Educação Infantil Conde do Pinhal. Por mais que nesse

caso pareça simples a solução, algumas escolas não tinham variações tão triviais. Para contornar esse caso, foi utilizado o *FuzzyWuzzy* para tokenizar as frases que representam os nomes das escolas, e, com base na cidade da aluna, obter a escola catalogada na planilha oficial da Secretaria do Estado de São Paulo (Governo do Estado de São Paulo, 2024), obtendo ainda informações homogêneas para o nome da instituição e o tipo de escola, preenchendo os valores nulos de alunas que não informaram se estudam em escolas públicas ou particulares.

A função utilizada contém alguns detalhes para otimizar o tempo de processamento, pois a biblioteca utilizada, apesar de eficiente, realiza processos que podem ser custosos ao software. O tempo final de execução da função, para as quase 600 linhas analisadas foi de 25 minutos. O código, apresentado no Apêndice A, na função *school_finder*, utiliza um esquema de *cache* em um dicionário em memória para evitar processamento repetido, além de filtros para otimizar a busca pelas escolas do estado. Um processo similar ao apresentado foi utilizado para normalizar as cidades inseridas.

O resultado final, após outras transformações, foi a Tabela 2 com 600 linhas.

Tabela 2 – Tabela final para análise gráfica das alunas participantes do programa.

Coluna	**Descrição**
index	Índice. Utilizado para operações com a tabela.
name	Nome da aluna. Utilizado principalmente para compara com outras tabelas e achar alunas recorrentes.
age	Idade da aluna. Calculado combinando a data de nascimento e a data de realização do evento.
city	Cidade da aluna. Normalizado através da função apresentada anteriormente.
state	Estado da aluna.
school_name	Nome da escola da aluna. Normalizado através da função apresentada anteriormente.
school_type	Enumerador tipo de escola. Particular ou Pública.
school_level	Enumerador nível escolar. Fundamental ou Ensino médio.
school_grade	Ano escolar da aluna. Fundamental de 1 a 9, médio de 1 a 3.
already_participated	Recorrência. Indicativo se a aluna participou de mais de uma edição do programa.
program_year	Ano do programa. Ano da edição que a aluna participou.

Fonte: O autor.

3.2.1.2 Análise dos dados coletados

Nesta seção, são apresentados os resultados obtidos a partir do conjunto de dados consolidado anteriormente, que inclui informações sobre as alunas que participaram do programa ao longo dos anos. São construídos gráficos e métricas com base no conjunto como um todo, e no *subset* a partir de 2021, ano em que a *Techschool* começou a ser realizado de forma remota, para analisar o principal fator a ser trabalhado: atingir o máximo de meninas em todo o Brasil, não

mais focado na cidade de São Carlos. São apresentados gráficos que mostram a distribuição das alunas por cidade, tipo de escola (pública ou particular), nível escolar (fundamental ou médio), e recorrência de participação no programa.

A utilização das bibliotecas *Matplotlib* e *Seaborn* foi fundamental para a criação dos gráficos apresentados. Essas ferramentas permitiram não apenas a visualização clara e eficaz dos dados, mas também a realização de análises estatísticas detalhadas. A importância dos gráficos na análise dos dados coletados não pode ser subestimada, pois eles fornecem uma maneira intuitiva e imediata de identificar tendências, *outliers* e padrões significativos.

Primeiramente, são mostrados os resultados obtidos sobre o conjunto de dados como um todo, e, posteriormente, sobre o subconjunto citado anteriormente. No total, foi possível obter dados de 577 alunas, que representa o total de alunas que passaram no programa em todas suas edições, desde 2018. Com uma média de mais de 82 alunas por edição, o programa contou com, na edição de 2024, 250 inscrições, das quais 70 foram selecionadas para participar. Na Figura 6 é ilustrado o gráfico de participantes por edição.

Figura 6 – Participantes separados por ano da *Techschool*. Note que a organização já realizou várias abordagens diferentes, tanto presencial quanto remotamente, e com números variados de alunas.

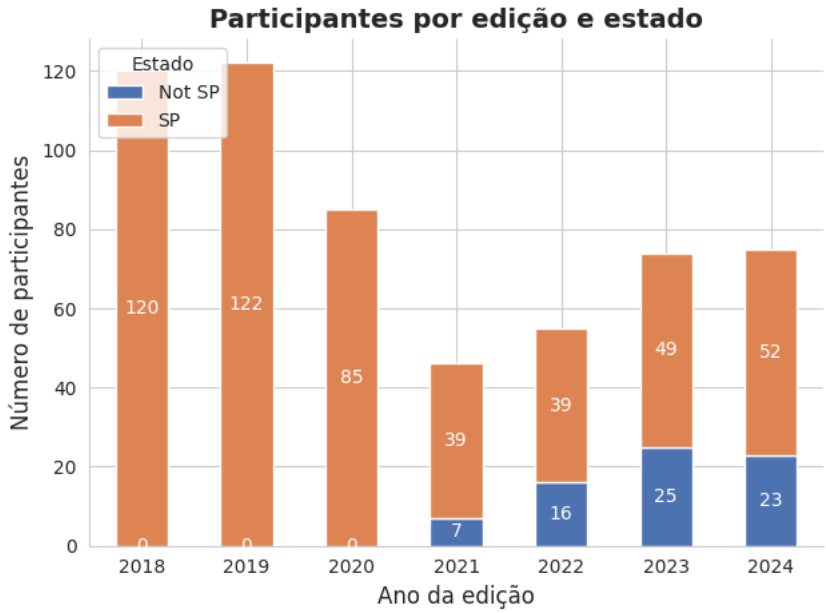
Fonte: O autor

Houve uma grande redução de alunas entre os anos 2019 e 2021, provavelmente influenciado pela pandemia, mas também para aumentar o nível de gerência e a qualidade de ensino do programa, visto que o desafio de engajar meninas, em sua grande parte entre 8 e 12 anos, em uma plataforma virtual é extremamente difícil, e pode necessitar de acompanhamento especial e ajuda parental.

Sequencialmente, foram realizadas algumas análises demográficas, com o objetivo de analisar o progresso da *Techschool*, e como a mesma vem se expandindo por todo o Brasil. Devido

às primeiras edições terem ocorrido de modo presencial, há uma quantidade significativamente maior de participantes do estado de São Paulo, tanto por serem as edições com o maior número de participantes, quanto focadas na cidade de São Carlos, em São Paulo. Na Figura 7 é ilustrado o gráfico de participantes por edição e estado.

Figura 7 – Participantes segregadas entre participantes do estado de São Paulo e de outros estados do Brasil.



Fonte: O autor

O crescente número de participantes fora do estado demonstra o ganho da realização do evento de forma remota, embora a proporção ainda se contrapõe, o que mostra que a *Techschool* ainda tem muito mais visibilidade no estado de São Paulo do que fora. No total, 71 alunas de outros estados já foram influenciadas, de dezoito estados diferentes, ilustrados na Figura 8.

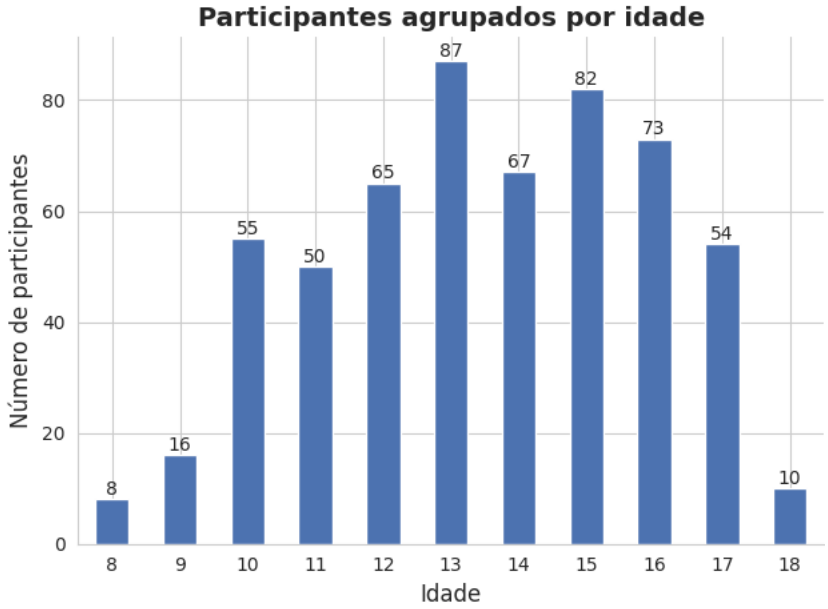
Um entendimento sobre a distribuição de idade das alunas é necessário para entender a homogeneidade do grupo e identificar padrões de engajamento em diferentes faixas etárias. Essa análise permite adaptar as estratégias pedagógicas de forma mais eficaz, garantindo que a *Techschool* atenda às necessidades específicas de cada faixa etária. Engajar jovens desde o ensino fundamental pode promover um interesse precoce em áreas tecnológicas e científicas, aumentando a probabilidade de continuidade nos estudos e carreiras nessas áreas. Além disso, ao monitorar a participação em diferentes idades, torna-se possível identificar o impacto da *Techschool* ao longo do tempo e ajustar intervenções para maximizar a inclusão e a motivação das alunas em todas as etapas educacionais. Na Figura 9 é ilustrado o gráfico com a idade das participantes.

Outro ponto interessante de se analisar é a distribuição entre alunas do ensino fundamental e médio, para entender como o programa está impactando diferentes etapas da educação básica. Essa distribuição revela a abrangência do programa e a sua capacidade de atrair e reter alunas em várias fases de sua formação escolar. Além disso, possibilita avaliar se há uma transição eficaz

Figura 8 – Participantes de estados fora de São Paulo. Note que há um boa dispersão entre as diferentes regiões do Brasil

Fonte: O autor

Figura 9 – Idade das participantes. Note como a mesma se aproxima de uma curva normal.

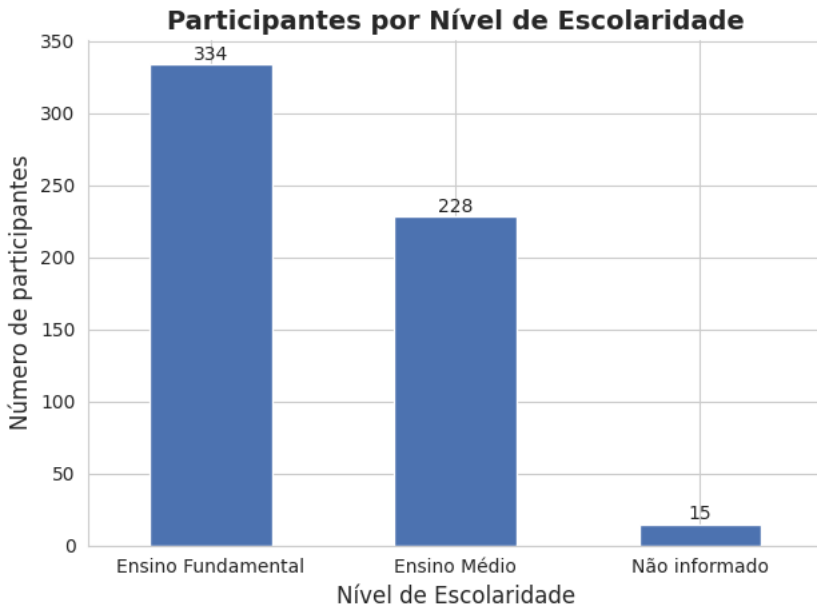


Fonte: O autor

de interesse e engajamento das alunas do fundamental para o médio, o que pode ser necessário para a sustentabilidade do interesse em áreas tecnológicas e científicas. Na Figura 10 é ilustrado o gráfico que agrupa as participantes por nível de escolaridade.

Ainda na linha de informações escolares, a análise da divisão entre alunas de escolas privadas e públicas no programa é essencial para compreender seu impacto social e sua contri-

Figura 10 – Participantes agrupadas por nível de escolaridade.

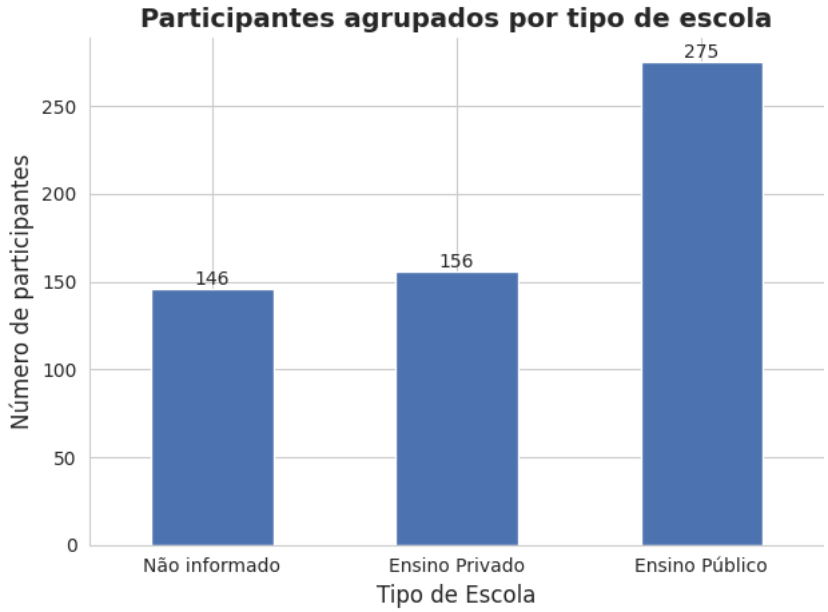


Fonte: O autor

buição para a redução das desigualdades educacionais. Isso permite avaliar se a *Techschool* está conseguindo alcançar uma diversidade socioeconômica significativa, promovendo a inclusão de alunas de diferentes contextos sociais. As alunas de escolas públicas, muitas vezes, enfrentam maiores desafios e menor acesso a recursos tecnológicos e educacionais avançados e, portanto, a participação dessas alunas no programa não só amplia suas oportunidades de aprendizado e desenvolvimento em áreas de tecnologia e inovação, como também contribui para a democratização do conhecimento, quebrando barreiras sociais e proporcionando um ambiente mais equitativo e incentivando a ascensão social por meio da educação de qualidade. Por outro lado, a participação de alunas de escolas privadas também é importante, pois promove a integração e o intercâmbio de experiências entre diferentes realidades, enriquecendo o aprendizado de todas as participantes e fomentando um ambiente mais diverso e inclusivo. Na Figura 11 é ilustrado o gráfico de participantes agregadas por tipo de escola.

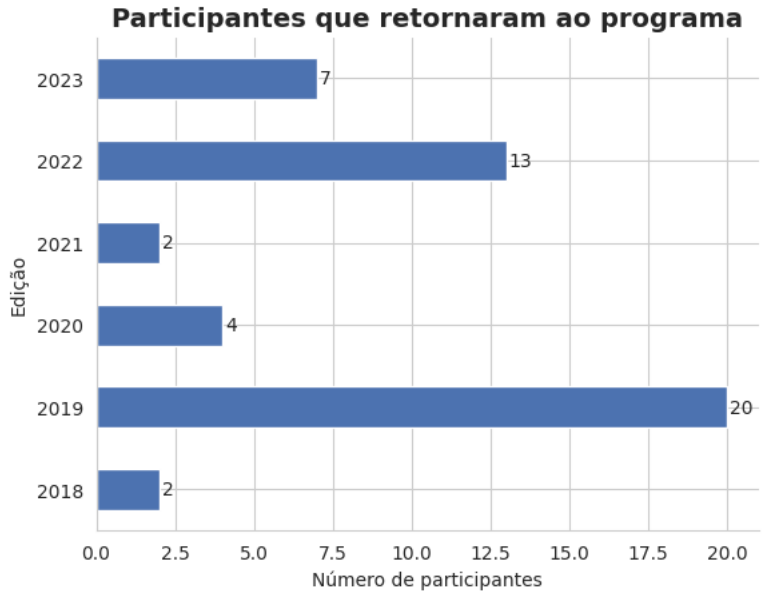
A última análise relevante a ser apresentada sobre as alunas nessa monografia é com relação à recorrência das mesmas, ou seja, alunas que retornaram e participaram de mais de uma edição. Esse indicador em especial revela o principal objetivo do programa, engajar meninas para as áreas de STEM pois indica o número de alunas que acharam pertinente participar de mais de uma edição. Vale ressaltar que esse indicativo, por mais que seja interessante, é subjetivo por duas razões: não considera alunas que se inscreveram novamente e foram recusadas na segunda edição por qualquer motivo, e também por, no conjunto de dados estudado, ter sido montado a partir da reincidência de nomes, o que é sensível a entradas de usuário. Na Figura 12 é ilustrado o gráfico de participantes em mais de uma edição.

Figura 11 – Participantes agrupadas por tipo de escola. O grande número de não informados se deve ao formulário menos incisivo nas edições iniciais, não cobrindo esse ponto, ainda mais que o mesmo tenha sido minimizado pelo uso do algoritmo de atribuição pelo nome de escolas iguais.



Fonte: O autor

Figura 12 – Participantes em mais de uma edição da *Techschool*.



Fonte: O autor

3.2.2 Análise técnica dos mentores

3.2.2.1 Coleta de dados e montagem do dataset

Assim como as alunas, foi possível montar um conjunto de dados referente aos mentores participantes da *Techschool*. Essa análise, além da utilidade dela por si só, indicando como o

programa se espalhou pelo Brasil e dados sobre a área de atuação dos mentores além do gênero, também foi uma etapa preparatória para o treinamento do modelo de seleção, que embora não utilize nenhum parâmetro como localidade ou gênero, utiliza a área de atuação profissional como parâmetro relevante na seleção do mentor. Para a análise de mentores, foram novamente cruzados dados de diferentes fontes e tecnologias para definir parâmetros úteis e palpáveis para os gráficos.

Diferente do formulário de inscrição de alunas, que haviam várias informações, a de mentores é enxuta e contém apenas algumas informações relevantes, como área de graduação e habilidades, e informações que são relevantes para a seleção, porém não analisadas, como disponibilidade. Isso forçou a obtenção de dados a partir de outras fontes, de acordo com o que foi possível obter a cada ano. No final, a análise ilustrada na Tabela 3 foi realizada apenas com os mentores para os quais obteve-se o endereço, para realizar análises demográficas cruzando, em sua maioria, uma planilha de apêndice que continha o endereço dos mesmos para o envio de camisetas do evento, para definir a cidade e estado dos mesmos a partir do CEP.

Tabela 3 – Relação de colunas por nível de utilidade nas planilhas da organização da *Techschool* para os mentores. Note que as colunas não são constantes entre as planilhas, podendo ter mais ou menos colunas a depender da edição.

Colunas irrelevantes	**Colunas Relevantes**	**Colunas úteis não utilizadas**
e-mail	Nome	Dias disponíveis
Número de telefone	Habilidades (lógica, design, etc)	Horas semanais
RG/ CPF	Área de graduação	-
Preferência monitoramento	-	-
Observações	-	-

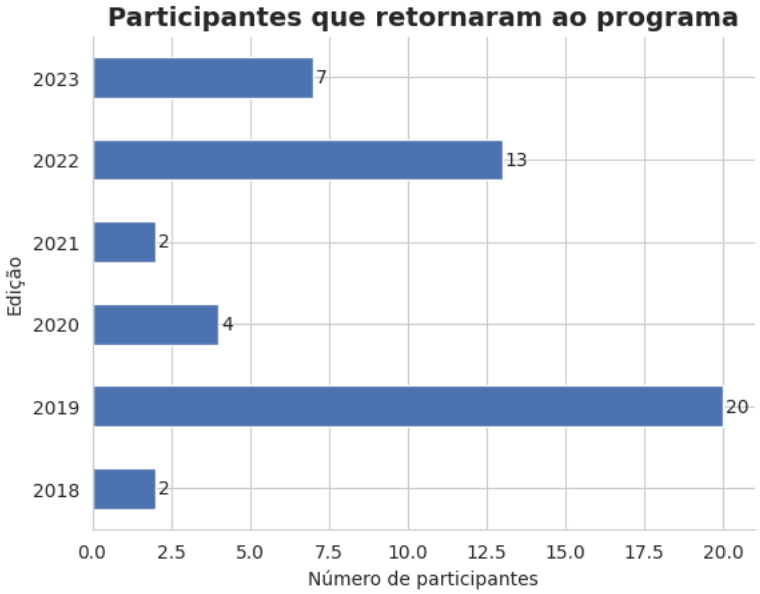
Fonte: O autor.

3.2.2.2 Análise dos dados coletados

Posterior à montagem do conjunto de dados, foram criados gráficos sobre os mentores que contribuíram para a *Techschool* ao longo de suas iterações. Os mentores têm um papel primordial no suporte técnico, emocional e motivacional oferecido às equipes de alunas participantes. Ao explorar a dispersão geográfica, especialidades e gênero, uma compreensão mais profunda da diversidade e dispersão da *Techschool* pode ser obtida. Novamente, a primeira parte é referente à localização geográfica, como ilustrado na Figura 13.

Outro dado relevante em relação aos mentores é o gênero, especialmente a presença de mentoras mulheres, servindo como modelos, encorajamento, e muitas vezes, um ente mais confortável para se identificar e se comunicar, criando algum laço no início do programa, uma etapa importante para o bom aproveitamento do conteúdo passado. Além disso, a presença de mentores de gêneros diversos não apenas enriquece a experiência de aprendizado das alunas, oferecendo uma variedade de perspectivas e experiências, mas também promove a equidade no

Figura 13 – Participantes que participaram mais de uma vez do programa.



Fonte: O autor

campo da tecnologia e inovação. Para obter tal dado, foi utilizado um recurso *online*, chamada *Gender Application Programming Interface* (API) (GENDER. . . , 2024), contando com uma base com quase sete milhões de nomes de 190 países diferentes, fornecendo dados confiáveis e com um bom embasamento estatístico para a maioria dos casos, e oferecendo iterações gratuitas para pequenos projetos.

Já sobre a área profissional, novamente era um campo livre, e, devido a isso, a entrada não padronizado dificulta a padronização e classificação das áreas de graduação. Para contornar esse problema, foi novamente utilizada a tokenização para modularizar as frases que compunham a descrição da graduação, e criado um *pipeline*, utilizando um algoritmo base de aprendizado de máquina, para vetorizar as sentenças, retirando *stop words* e palavras que podem influenciar o algoritmo, e, alimentado com um vocabulário de mais de mil linhas contendo exemplos de palavras e as suas respectivas áreas, de modo que foi possível classificar a área de graduação com uma precisão de 85%. O código é apresentado no Apêndice A, como natural *language classifier*, e foi utilizado em múltiplas ocasiões de processamento de linguagem natural durante o desenvolvimento deste trabalho.

Por fim, a Tabela 4 ilustra os 205 mentores, a partir da qual foi possível extrair dados confiáveis. Vale ressaltar que antes de 2021, o programa era realizado presencialmente, logo toda a parte de análise demográfica fica totalmente enviesada. Logo, para este caso em específico, optou-se por abordar apenas os mentores a partir de 2021, uma vez que apresentam dados confiáveis.

Tabela 4 – Tabela final para análise gráfica dos mentores participantes do programa.

Coluna	Descrição
index	Índice. Utilizado para operações com a tabela.
name	Nome do mentor. Utilizado principalmente para comparar com outras tabelas para fornecer endereços e definir o gênero.
graduation	Área de graduação do mentor, fornecido em formulários de inscrição.
city	Cidade do mentor. Obtido a partir do CEP de entrega da camiseta do evento.
state	Estado do mentor.
program_year	Ano de participação.
skills	Habilidades do mentor. Fornecido a partir de 2022. Conta com enumeradores como Lógica de programação, Ideação, Prototipação ou Design *Thinking*, Oratória, Comunicação e *Pitch*, Plano de Negócios.

Fonte: O autor.

3.2.2.3 Análise dos dados coletados

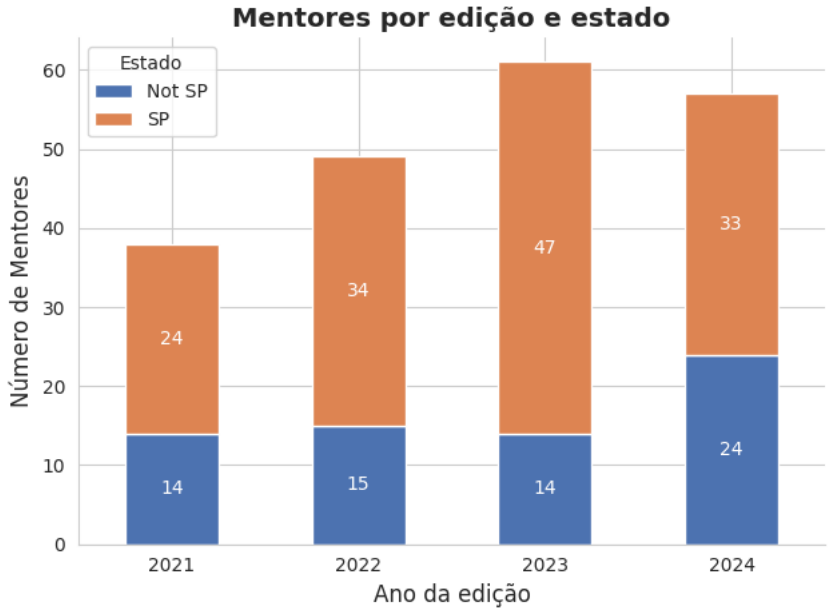
Paralelamente ao feito para as alunas, foi realizado um pequeno estudo a cerca dos dados coletados sobre os mentores. Os mentores têm um papel primordial no suporte técnico, emocional e motivacional oferecido às equipes de alunas participantes. Ao explorar a dispersão geográfica, especialidades e frequência de envolvimento desses mentores, tem-se como objetivo obter uma compreensão mais profunda da diversidade e impacto de seu auxílio. Inicialmente, foi investigado como a *Techschool* se adaptou para abranger uma variedade mais ampla de mentores em todo o Brasil, expandindo para além de São Carlos. Na Figura 14 é ilustrado se o mentor localiza-se no estado de São Paulo ou não.

Novamente, é importante ressaltar que, diferente das alunas, esses dados não representam exatamente todos os mentores que participaram do programa, nem mesmo todos que participaram das edições citadas. De qualquer modo, é possível verificar um leve crescimento na quantidade de mentores de outros estados, como ilustrado na Figura **??**.

Abordando a área profissional, é possível notar uma grande preferência pela área da informática, que é, naturalmente, o foco do programa. Entretanto, a presença de mentores multidisciplinares é de suma importância para o desenvolvimento integral dos grupos. Um produto de tecnologia não se resume apenas à programação, a compreensão das regras de negócio, das necessidades dos clientes e os conceitos de *design* são igualmente importantes para garantir a qualidade e a eficácia do produto final (JURISTO; MORENO; SANCHEZ-SEGURA, 2007).

Focando no grupo de alunas, essa mesma multidisciplinaridade não apenas enriquece o processo de desenvolvimento, mas também melhora a gestão das equipes, incentiva o pensamento

Figura 14 – Mentores diferenciados pelos residentes e não residentes no estado de São Paulo.



Fonte: O autor

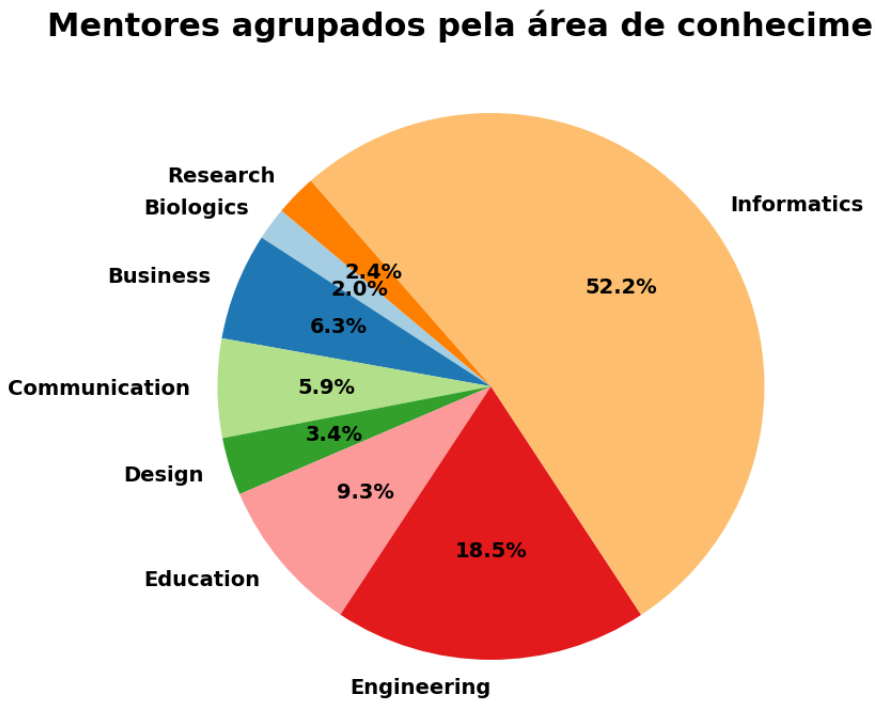
Figura 15 – Mentores diferenciados pelos residentes e não residentes no estado de São Paulo.

Fonte: O autor

criativo e diversifica os métodos de ensino e a procura por soluções. Essa diversidade de conhecimentos e habilidades aumenta a eficiência e a abrangência do programa, proporcionando uma experiência educacional mais rica e abrangente para as meninas. Na Figura 16 são ilustrados os mentores por sua área de atuação.

Por último, ao reparar no gênero dos mentores, é possível perceber que a *Techschool* cumpre o esperado, apresentando uma maior quantidade de mentoras. Vale reforçar que essa

Figura 16 – Mentores diferenciados pela sua área de atuação, atribuída pelo algoritmo de aprendizado de máquina.



Fonte: O autor

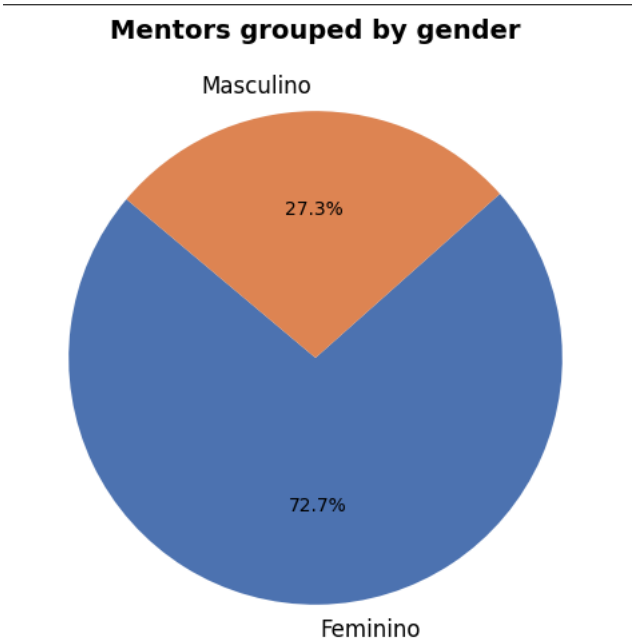
proporção é um detalhe importante na jornada das alunas, auxiliando tanto no desenvolvimento e abertura delas durante a *Techschool*, quanto na busca por inspirações e exemplos de sucesso na área da tecnologia. Na Figura 17 são ilustrados os mentores agregados por gênero.

3.3 Modelo de seleção de mentores

O sucesso da *Techschool* depende fortemente da qualidade e diversidade dos mentores envolvidos. Os mentores desempenham um papel de suma importância no desenvolvimento técnico e pessoal das alunas, fornecendo suporte, orientação e inspiração ao longo do processo. Portanto, o modelo de seleção de mentores é um componente essencial da *Techschool*, garantindo que as alunas recebam a melhor orientação possível. Infelizmente, nos últimos anos, o processo de seleção de mentores vem se tornando algo custoso. Para comparação, a edição de 2024 contou com 439 inscrições, das quais menos de 70 foram selecionados para participar do projeto, e, conforme o mesmo cresce e amadurece, a capacidade técnica, e qualidade de ensino dos mentores deve acompanhar.

A seleção de mentores envolve vários critérios que visam maximizar a eficácia da *Techschool*. Esses critérios incluem a experiência profissional dos mentores, sua capacidade de comunicação, o entendimento das necessidades do programa e disponibilidade. Além disso, a diversidade é um fator chave, com a inclusão de mentores de diferentes regiões, gêneros e áreas de conhecimento para fornecer uma abordagem multidisciplinar ao ensino de tecnologia.

Figura 17 – Mentores agrupados pelo gênero. Vale destacar que essa atribuição foi realizada por um software.



Fonte: O autor

Dessa forma, a análise detalhada do perfil dos mentores e a aplicação de critérios rigorosos de seleção são fundamentais para assegurar que a *Techschool* continue a ser um ambiente enriquecedor e eficaz para o desenvolvimento das futuras líderes em tecnologia.

O treinamento do modelo envolveu duas etapas, que são detalhadas nesta seção, a coleta de dados, e o treinamento e refino do modelo. Um ponto importante de enfatizar é que o desenvolvimento do modelo foi, inicialmente, apenas para triagem, pois, como abordado a seguir, algumas limitações técnicas impedem a criação de um modelo de extrema confiabilidade, como falta de dados, além da possibilidade de um viés de seleção (WINSHIP; MARE, 1992), pois candidatos igualmente capacitados aos selecionados, que foram recusados pela limitação de vagas, o que pode causar um *underfitting*, ou seja, devido ao viés, o modelo pode não se adaptar bem a dados e condições muito restritivas.

3.3.1 Algoritmo de webscraping

Infelizmente, a descrição redigida pelos mentores é muito variável, e qualquer tentativa de análise desse dado pode valorizar indivíduos que escrevam mais ou menos, de acordo com a moda do conjunto de dados, além de ser mais facilmente influenciável por palavras-chave. Para superar esse obstáculo, foi elaborado um *webscraper*, que acessa o perfil profissional dos mentores no *LinkedIn*, e extrai informações relevantes para a seleção, como experiência profissional e educacional, voluntariado, certificações e habilidades.

Entretanto essa metodologia tem duas fragilidades, que podem comprometer a confiabili-

dade do modelo. A primeira, devido ao tempo, os perfis dos candidatos mais antigos pode não refletir mais a condição dos mesmos quando foram selecionados ou desqualificados, significando que quanto mais antiga a edição utilizada no treinamento, menos confiáveis são os dados. A segunda, se refere a uma limitação social, que nem todos os candidatos possuem ou enviaram seu perfil do *LinkedIn* para a organização, o que reduziu a quantidade de dados disponíveis ainda mais.

Com essas observações, optou-se por realizar a contextualização do modelo utilizando os dados apenas dos últimos dois anos. O algoritmo é simples; recebe de entrada uma lista de perfis no *LinkedIn*, acessa um por um, na página do candidato, extrai as informações relvantes. Isso é feito usando o chamado *XML Path Language* (XPATH) para identificar os elementos desejados nas páginas, ilustrado na Figura 18. O XPATH é basicamente um caminho dentro da estrutura de tags *Hyper Text Markup Language* (HTML) do site, exemplificada abaixo, que leva até o texto que queremos extrair, e, embora não seja o método mais eficiente de webscraping, é o mais simples de ser aplicado (GUNAWAN *et al.*, 2019). Os dados foram salvos em memória, em um *DataFrame* Pandas, e, após a conclusão, convertidos para um formato *.csv*.

Figura 18 – Exemplo simplificado de HTML de um site. Para acessar o valor "Sidney", por, exemplo, o XPATH seria /html/body/table/tr[3]/td[1].

```
<html>
 <body>
  <span class="heading">
   Weather
   forecast
  </span>
  <table>
   <tr>
    <th>City</th>
    <th>Today</th>
    <th>Tomorrow</th>
   </tr>
   <tr>
    <td>Melbourne</td>
    <td>20</td>
    <td>22</td>
   </tr>
   <tr>
    <td>Sydney</td>
    <td>25</td>
    <td>22</td>
   </tr>
   <tr>
    <td>Adelaide</td>
    <td>26</td>
    <td>26</td>
   </tr>
  </table>
  <span></span>
 </body>
</html>
```

Fonte: (PENMAN; BALDWIN; MARTINEZ, 2009)

Após concluir a varredura de todos os mentores que disponibilizaram o perfil, foi obtido um *.csv* com 590 linhas de informações de mentores inscritos, que foi importada para o escopo do modelo e convertida novamente em um *DataFrame* Pandas. A base contém o nome do mentor,

se o mesmo foi selecionado ou não para o programa, e as informações do *webscraper*, que foram obtidas no formato de listas, por exemplo, se uma pessoa tem três empregos listados em seu perfil, o algoritmo retornou uma estrutura de lista como ["cargo 1, data, empresa", "cargo 2, data, empresa", "cargo 3, data, empresa"]. As informações coletadas foram: informações profissionais e cargos, informações educacionais, cursos e certificações concluídas, habilidades listadas, e por último voluntariado realizado.

Todas essas informações precisaram ser tratadas. Para a lista de experiências profissionais, por exemplo, foi aplicado um algoritmo similar, porém mais refinado do que o apresentado de tokenização utilizado anteriormente. Essa versão classificou os mentores em áreas profissionais como educação, tecnologia, negócios, e assim por diante. Similar às experiências profissionais, a educação também foi transformada. Nesse caso, primeiro era validado se o candidato realizou alguma graduação, e então, caso o curso realmente seja de uma faculdade, o classifica em categorias como matemática aplicada e tecnologia, engenharia e construção, ciências sociais e direito, entre outros.

Sobre os cursos, certificados e habilidades, foi realizada apenas uma contagem, devido à complexidade e falta de estruturação e padronização dos dados coletados. Por último, sobre o voluntariado, foi criada uma coluna *has_volunteer*, indicando apenas se a pessoa já realizou algum tipo de voluntariado ou não. Vale ressaltar que alguns outros dados foram coletados e estruturados, porém, devido à sua baixa ou nenhuma correlação com o alvo, após breve análise de características e correlação, como anos de experiência, tipo de faculdade, e outras, não foram utilizados.

Os dados do *webscraping* foram então combinados com os dados de disponibilidade dos mentores, como quantidade de dias de reuniões síncronas eles têm disponíveis, e quantidade de horas semanais, pois os mesmos foram considerados igualmente relevantes para a seleção. A Tabela 5 ilustra os dados obtidos ao final do modelo de triagem.

Uma última etapa de pré-processamento necessária para o modelo, é a conversão de variáveis categóricas para booleanas. Isso foi feito simplesmente aplicando o método *get_dummies()* do Pandas, que identifica as categorias possíveis, e gera uma coluna nova para cada uma delas. Essa metodologia foi aplicada tanto na área profissional quanto na área de graduação dos candidatos.

Após essa última etapa, foi utilizada a função *train_test_split* da biblioteca *scikit-learn* do Python para separar os dados, de forma aleatória e em uma proporção pré-definida, em conjunto de treino e de teste, com a respectiva proporção de 80% treino e 20% teste, para posterior avaliação do modelo.

Tabela 5 – Tabela final para contextualização e treinamento do modelo de triagem.

Coluna	Descrição
index	Índice. Utilizado para operações com a tabela.
name	Nome do mentor.
target	Booleano. Indica se o mentor foi ou não selecionado.
disponibility_days	Numérico (1 - 5). Quantidade de dias que o mentor se disse disponível na inscrição.
weekly_hours	Numérico. Quantidades de horas semanais o mentor se disse disponível na inscrição.
is_volunteer	Booleano. Indica se o mentor já realizou outro projeto de voluntariado.
area	Categórico. Indica a área de atuação profissional atribuida ao mentor.
graduation_area	Categórico. Indica a área de graduação atribuída ao mentor.
more_than_one_grad	Booleano. Indica se o mentor realizou mais de uma graduação.
m*ore_than_3_certs*	Booleano. Indica se o mentor realizou mais de três certificações.
more_than_4_skills	Booleano. Indica se o mentor listou mais de 4 skills relevantes ao programa, como relacionadas à programação, ensino, negócios.

O autor.

3.3.2 Seleção e refino do modelo

A última etapa, igualmente importante, é o treinamento do modelo. Para esse caso, ao invés de construir um modelo do zero, o que seria extremamente trabalhoso e complexo, optou-se por utilizar modelos pré-estruturados, que seguem algoritmos conhecidos, baseados em pesquisas bem consolidadas, e otimizadas. A biblioteca *scikit-learn* (SCIKIT-LEARN. . . , 2024) oferece várias opções de modelos.

O primeiro passo foi realizar um teste com vários dos modelos tipicamente utilizados em problemas de classificação, dentre os quais, vale destacar o *DecisionTreeClassifier*, *AdaBoostClassifier*, *KNeighborsClassifier*, e o *RandomForestClassifier*. Cada um desses modelos, suportados pela biblioteca, são uma base para um projeto simples, e são pré-configurados para fácil utilização, entretanto, simbolizam processos completamente diferentes de classificação, e tem seus prós e contras definidos pela comunidade. Por ser um projeto básico de classificação binária, dois modelos se destacam, embora todos os listados foram testados quanto à performance, sendo os destaques o *Ada Boost* e o *Decision Tree*, pois ambos lidam bem com esse tipo específico de problema.

Um recurso utilizado foi a utilização dos chamados pesos, que servem para indicar ou balancear o modelo sobre as decisões que o mesmo pode tomar, tendo em vista que os dados coletados apresentam ocorrências de recusa (0) muito maiores que de aceite (1). Somente esse

recurso já auxiliou, e muito, no entendimento do modelo sobre o conjunto de dados, aumentando o *recall* (as ocorrências) de aceite, que em alguns modelos chegava a 5%, indicando o modelo recusaria 95% dos candidatos.

Um último recurso utilizado foi o *grid search*. O *grid search* consiste em uma metodologia de aumento de eficiência do modelo ao alterar os chamados hiperparâmetros (LIASHCHYNSKYI; LIASHCHYNSKYI, 2019). Seu uso, basicamente, instancia o modelo e realiza várias iterações de treino e teste, com um pontuador pré-definido, e então a função retorna a melhor combinação de hiperparâmetros para o caso abordado. Os hiperparâmetros, por sua vez, são as configurações personalizáveis dos modelos pré construídos, por exemplo, é possível alterar a máxima profundidade do algoritmo de *Decision Tree*, que, embora não será detalhado nessa seção, está diretamente ligado ao tamanho e refino do modelo. O *Grid Search* não é o único meio de manipular os hiperparâmetros, há também vários outros, como o *Random Search* (ANDRADÓTTIR, 2006), porém como o contexto é limitado e sem um requisito de eficiência relevante, o *Grid Search* foi utilizado.

Para utilização do G*rid search*, foram selecionados alguns parâmetros para realizar a busca pela melhor combinação, como apresentado na Tabela 6. O primeiro modelo foi o *Decision Tree Classifier*, que é um algoritmo de aprendizado supervisionado comumente usado para tarefas de classificação. Ele funciona construindo uma árvore de decisão a partir do conjunto de dados de treinamento, onde cada nó da árvore representa uma decisão baseada em um atributo. O objetivo é dividir o conjunto de dados em subconjuntos cada vez mais homogêneos em relação à variável alvo. Durante o treinamento, a árvore é expandida até que uma condição de parada seja atingida, como uma profundidade máxima pré-definida ou quando todos os exemplos de treinamento pertencem à mesma classe.

O *Ada Boost*, ou *Adaptive Boosting*, é um algoritmo de aprendizado de máquina que constrói um modelo forte a partir de uma sequência de modelos fracos. Em cada iteração do processo de treinamento, o *Ada Boost* ajusta pesos aos exemplos de treinamento, de forma que os exemplos mal classificados recebam mais peso para as iterações subsequentes. Isso permite que o algoritmo concentre sua atenção nos exemplos mais difíceis de classificar. Os modelos fracos, geralmente árvores de decisão rasas, são combinados de forma ponderada para formar um modelo final robusto. O *Ada Boost* é amplamente utilizado devido à sua eficácia em lidar com conjuntos de dados desbalanceados e à sua capacidade de generalização para novos dados. Na Tabela 7 são ilustrados os hiperparâmetos do *Ada Boost*.

Para avaliar o desempenho do modelo, a base de pontuação escolhida foi o *F1-Score*, definido matematicamente como a média harmônica entre a precisão do modelo e a revocação (*recall*). A precisão é a proporção de verdadeiros positivos sobre todos os exemplos que foram classificados como positivos, e indica a precisão dos positivos previstos pelo modelo. A revocação consiste na proporção de verdadeiros positivos encontrados pelo modelo sobre todos os verdadeiros positivos e os falsos negativos, e indica a capacidade do modelo de encontrar todos

Tabela 6 – Hiperparâmetros de uma decision tree escolhidos para o grid search (SCIKIT-LEARN. . . , 2024).

Coluna	Descrição	Valores
class_weight	Balanceamento de classes, utilizado para casos que uma classe tem mais ocorrências que outra, podendo prejudicar as predições.	None, 'balanced'
criterion	Formulações matemáticas para medir a eficiência da divisão da árvore.	'gini', 'entropy'
max_depth	Profundidade da árvore.	None, 10, 20, 30, 100
max_features	Quantidade de *features* utilizadas na procura pela melhor divisão.	None, 'sqrt', 'log2'
max_leaf_nodes	Quantidade máxima de nós na árvore.	None, 10, 50, 100
min_impurity_decrease	Critério para divisão de um nó impuro.	0, 3, 5
min_samples_leaf	Número mínimo de amostras que deve estar presente em um nó folha para que ele seja dividido.	0.5, 2, 4, 7, 10
min_samples_split	Número mínimo de amostras necessárias em um nó para que ele seja considerado para divisão.	0.5, 2, 4, 7, 10
splitter	Estratégia usada para dividir cada nó.	'best', 'random'

Fonte: O autor.

Tabela 7 – Hiperparâmetros de um modelo de Ada Boost escolhidos para o grid search.

Coluna	Descrição	Valores
estimator	Classificadores fracos a ser usado como base.	Decision Tree Classifier(max_depth=2, 3 OU 4)
n_estimators	número de "estimators"que serão treinados sequencialmente.	10, 30, 70, 100, 150
learning_rate	Contribuição de cada modelo para o modelo final.	0.1, 1.0, 2, 5
algorithm	Especificação de qual algoritmo específico deve ser usado para treinar o modelo.	'SAMME', 'SAMME.R'

Fonte: (SCIKIT-LEARN. . . , 2024).

os positivos. Por fim, o *F1-Score* é a média harmônica dos dois, por apresentar um equilíbrio e a penalização de casos extremos (precisão muito baixa, por exemplo). O código utilizado está exemplificado no Apêndice A.

É importante ressaltar que, dependendo do modelo e do objetivo, a utilização de uma métrica ou outra pode variar. Por exemplo, em um modelo de previsão de doenças cardíacas, um falso negativo é bem mais impactante do que um falso positivo, pois um falso negativo significa deixar de identificar uma pessoa doente, o que pode ter consequências graves. Nesse caso, utilizar a revocação como métrica pode ser recomendado, pois pretende-se apresentar o maior número

de verdadeiros positivos. Enquanto isso, no caso de algoritmos de indicação de vídeos, por exemplo, o impacto de mostrar vídeos que não agradam ao usuário (falsos positivos) pode ser mais prejudicial do que perder alguns vídeos que o usuário gostaria (falsos negativos). Assim, a precisão se torna mais atrativa, pois o intuito é minimizar os falsos positivos. Por fim, o *F1-Score* é recomendado para o caso de estudo atual, onde os dados se encontram desbalanceados (muito mais ocorrências de um caso - no modelo atual, 483 casos de recusa, contra 107 de aceite), pois, nesses casos, o enfoque em uma alta precisão, mas que indique todos como falsos (80 % de precisão) não é algo realmente efetivo.

Para a visualização dos dados, foram utilizados dois artifícios; o reporte de classificação, fornecido pela biblioteca *scikit-learn*, é um método prático de obter os principais dados estatísticos da previsão, como precisão, revocação, *F1-Score*, acurácia, e outros dados relevantes; e a matriz de confusão, uma ferramenta fundamental para a avaliação de modelos de classificação, que fornece uma visão detalhada do desempenho do modelo, mostrando a quantidade de previsões corretas e incorretas, distribuídas entre as diferentes classes. A matriz de confusão ajuda a identificar não apenas a acurácia geral do modelo, mas também os tipos de erros que ele comete. No caso atual, um modelo de triagem apenas, a presença de falsos positivos não é tão prejudicial, sendo assim, a meta é obter uma matriz de precisão onde hajam poucos ou nenhum falso negativo, e a maior quantidade possível de verdadeiros negativos e de verdadeiros positivos, mantendo como métrica um bom *F1-Score*.

Os resultados do modelo de *Decision Tree* são ilustrados na Figura 19. Note que, novamente, o intuito do modelo não é definir a combinação perfeita dos candidatos, os dados coletados para tal são insuficientes, mas sim definir quais não apresentam características relevantes para participar. A melhor combinação dos hiperparâmetros encontrada foi a seguinte: class_weight: None; criterion: 'gini'; max_depth: 100; max_features: 'log2'; max_leaf_nodes: 50; min_impurity_decrease: 0; min_samples_leaf: 10; min_samples_split: 7; min_weight_fraction_leaf: 0.1; splitter: 'random'. A matris de confusão é ilustrada na Figura 20.

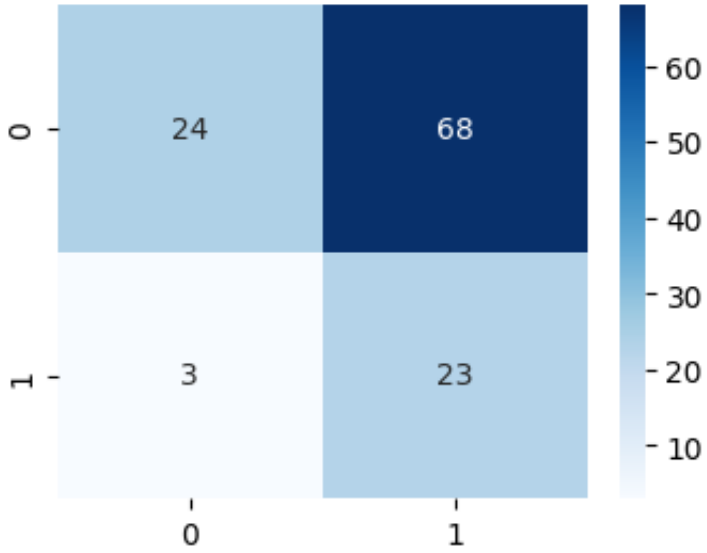
Figura 19 – *Classification Report* gerado do melhor modelo encontrado para os dados utilizando o algoritmo de *Decision Tree*.

	precision	recall	f1-score	support
0	0.81	0.38	0.52	92
1	0.24	0.69	0.36	26
accuracy			0.45	118
macro avg	0.53	0.54	0.44	118
weighted avg	0.69	0.45	0.48	118

Fonte: O autor

Já o *Ada Boost*, utilizando várias *Decision Trees* como modelo fraco, obteve o seguinte resultado: algorithm: 'SAMME'; estimator: DecisionTreeClassifier(max_depth=3); learning_rate: 1.0; n_estimators: 30, conforme ilustrado na Figura 21. A matriz de confusão é ilustrada na Figura 22.

Figura 20 – Matriz de confusão gerada do melhor modelo encontrado para os dados utilizando o algoritmo de *Decision Tree*. Note que o número de falsos negativos foi extremamente baixo, e que o modelo já conseguiu aliviar a carga da busca em 20&



Fonte: O autor

Figura 21 – *Classification Report* gerado do melhor modelo encontrado para os dados utilizando o algoritmo de Ada Boost.

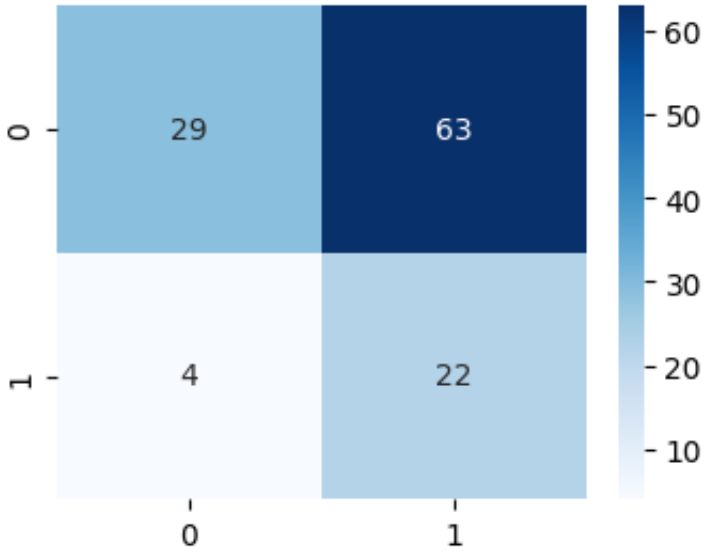
	precision	recall	f1-score	support
0	0.80	0.74	0.77	92
1	0.27	0.35	0.31	26
accuracy			0.65	118
macro avg	0.54	0.54	0.54	118
weighted avg	0.68	0.65	0.67	118

Fonte: Elaborada pelo autor

3.4 Considerações finais

A seção de desenvolvimento detalhou a análise dos dados coletados das alunas e dos mentores da *TechSchool* e revelou percepções valiosas sobre o impacto da *TechSchool* na formação das participantes e a eficiência dos mentores. A análise das alunas demonstrou um progresso no desenvolvimento do programa, destacando padrões de crescimento que podem orientar futuras edições da *TechSchool* a se expandirem. A avaliação dos mentores permitiu identificar características preferíveis na seleção de mentores, e demonstrou que o programa apresenta um valor e desenvolvimento social e pessoal não só às alunas, mas a todos os envolvido. Além disso, a aplicação de algoritmos de aprendizado de máquina foi importante para processar e interpretar os dados complexos, fornecendo previsões identificando correlações significativas para um modelo final de triagem de candidatos. Este processo não só validou a eficácia da *TechSchool* como também estabeleceu uma base sólida para melhorias contínuas e expansão estratégica do programa, contribuindo para a capacitação e empoderamento de jovens mulheres

Figura 22 – Matriz de confusão gerada do melhor modelo encontrado para os dados utilizando o algoritmo de *Ada Boost*. Note que o número de falsos negativos foi extremamente baixo, e que o modelo já conseguiu aliviar a carga da busca em 20&



Fonte: Elaborada pelo autor

na área de STEM.

No próximo capítulo são apresentadas as conclusões obtidas com a execução deste projeto de pesquisa.

Capítulo 4

CONCLUSÃO

4.1 Contribuições

Essa monografia tinha como objetivo três principais frentes; a análise gráfica das alunas participantes que proporcionou percepções valiosas sobre a distribuição demográfica, recorrência de participação e características socioeconômicas das alunas ao longo dos anos. Essas visualizações ajudaram a compreender melhor o impacto da *TechSchool* e identificar áreas de melhoria para futuras edições. Da mesma forma, a análise gráfica dos mentores permitiu uma avaliação abrangente da diversidade de gênero e áreas de conhecimento entre os mentores participantes. Essas informações foram essenciais para garantir uma representação equilibrada e inclusiva no suporte às alunas. Além disso, o modelo de seleção de mentores desenvolvido ofereceu uma abordagem sistemática e objetiva para identificar os mentores mais adequados com base em critérios específicos, promovendo assim uma experiência de mentoria mais eficaz e personalizada para as participantes do programa.

É incontestável que a maior contribuição do trabalho apresentado é para o futuro da *TechSchool*, mostrando dados sobre o passado da mesma, pretende-se ajudar a estruturar e planejar melhor os próximos passos, para que assim a *TechSchool* ajude em seu objetivo de inserção de meninas e o despertar de interesse das jovens na área da tecnologia. Para o autor desta monografia, o trabalho ajudou profundamente no entendimento do processo de coleta e tratamento de dados, principalmente em como estruturar e analisar a confiabilidade de um conjunto de dados, além de melhorar o entendimento sobre o funcionamento, e principalmente a avaliação de modelos de aprendizado de máquina.

4.2 Relacionamento entre o Curso e o Projeto

A realização dessa pesquisa utilizou diversos ensinamentos e habilidades desenvolvidos ao longo do curso de Engenharia de Materiais e Manufatura. Dentre as principais disciplinas é possível citar, Métodos numéricos e computacionais I e II, que introduziram os alunos à área da tecnologia; Sistemas de informação, que auxilio no entendimento de estruturas de dados e como tratá-los; Estatística Aplicada à Engenharia, utilizada durante toda a pesquisa; e, por último, disciplinas do intercâmbio, como *Recent Advances in Intelligent Engineering*, que introduziu o aluno à área de aprendizado de máquina. Além disso, habilidades de comunicação e trabalho em

grupo desenvolvidas ao longo de toda a graduação foram essenciais para a participação do aluno como mentor do programa, em sua edição mais recente em 2024.

4.3 Considerações sobre o Curso de Graduação

O curso de Engenharia de Materiais e Manufatura, da Escola de Engenharia de São Carlos (EESC) prepara o aluno para o mercado de trabalho, oferecendo uma boa base teórica sobre o que o curso propõe, e abrindo portas para seus alunos em várias frentes, como pesquisa e indústria. Além disso, gera valor ao apresentar aos alunos, e permitir o uso em diversos trabalhos, de ferramentas computacionais e de simulação, liderando o mesmo a um mundo completamente diferente da engenharia tradicional, e muito promissor para o futuro. O curso propôs um bom equilíbrio entre teoria e prática, oferecendo disciplinas extremamente teóricas, e disciplinas extremamente práticas e com laboratório para embasar qualquer afirmação e suportar qualquer pesquisa que o aluno possa fazer, além de oferecer disciplinas de design e desenvolvimento de produto, de extremo valor para o mercado de trabalho.

Uma consideração que gostaria de fazer sobre o curso, é que, por mais que os alunos sejam permitidos a usar softwares, acredito que ainda falta certo incentivo e iniciativa por parte da organização para engajá-los nas disciplinas envolvidas demonstrando os benefícios que o uso dos mesmos podem trazer. Além disso, uma revisão na maneira de como esse tipo de conteúdo é passado aos alunos, e a qualidade e atualidade do conteúdo devem ser levados em conta, um exemplo disso é a disciplina Engenharia Auxiliada Por Computador, que se modernizou, na área teórica, ao utilizar Python ao invés de meios tradicionais para o cálculo de estruturas em suas aulas teóricas, embora tenha ainda pontos a serem discutidos na parte prática.

Os resultados esperados desta pesquisa envolvem a identificação de áreas de melhoria e expansão do programa, bem como dados que evidenciem seu impacto positivo na trajetória acadêmica e profissional das participantes. Por fim, este estudo visa reiterar a importância de programas educacionais direcionados a meninas no campo de STEM, destacando a necessidade de investimentos e políticas públicas que promovam a igualdade de gênero nesse contexto, impulsionando o direcionamento futuro do programa, a avaliação de seu impacto e o avanço de programas educacionais inclusivos.

4.4 Limitações e Trabalhos Futuros

Essa última seção da monografia foi dedicada a dois tópicos correlacionados: limitações do trabalho atual, e possibilidades de melhoria e implementações do trabalho no futuro. Sobre as limitações, deve-se evidenciar a efemeridade dos dados, por exemplo, dos mentores, que ao serem buscados no dia de hoje, referente à mentores de dois anos atrás, podem apresentar certo viés. Outro ponto é a falta de dados, por exemplo, de mentores, como idade e gênero, que podem

ser perguntados nos formulários de inscrição, além da implementação de uma base unificada de dados sobre os participantes de cada edição, facilitando assim trabalhos futuros e criando um *pipeline* para alimentar possíveis futuras implementações de análise de dados e aprendizado de máquina utilizando o programa como alvo. Sobre os trabalhos futuros e direcionamentos que o programa pode seguir, baseado no conteúdo apresentado, pode-se enfatizar:

- implementação de uma base de dados sobre participantes do evento e reestruturação do formulário;
- Refino do modelo treinado apresentado, utilizando dados mais confiáveis e frescos para alimentá-lo;
- Expansão das análises, conforme o programa cresce e também se expande, aumentando os dados atuais e gerando novos.

REFERÊNCIAS

ANDRADÓTTIR, S. An overview of simulation optimization via random search. **Handbooks in operations research and management science**, Elsevier, v. 13, p. 617–631, 2006. Citado na página 56.

BABBIE, E. R. **The practice of social research**. [S.l.]: Cengage AU, 2020. Citado na página 25.

BEDDOES, K.; PANTHER, G. Gender and teamwork: An analysis of professors' perspectives and practices. **European Journal of Engineering Education**, Taylor & Francis, v. 43, n. 3, p. 330–343, 2018. Citado na página 21.

BELLAZZI, R.; ZUPAN, B. Predictive data mining in clinical medicine: current issues and guidelines. **International journal of medical informatics**, Elsevier, v. 77, n. 2, p. 81–97, 2008. Citado na página 23.

BIRD, S.; KLEIN, E.; LOPER, E. **Natural language processing with Python: analyzing text with the natural language toolkit**. [S.l.]: "O'Reilly Media, Inc.", 2009. Citado na página 37.

BISHOP, C. M. Pattern recognition and machine learning. **Springer google schola**, v. 2, p. 1122–1128, 2006. Citado na página 34.

DASGUPTA, N.; STOUT, J. G. Girls and women in science, technology, engineering, and mathematics: Steming the tide and broadening participation in stem careers. **Policy Insights from the Behavioral and Brain Sciences**, SAGE Publications Sage CA: Los Angeles, CA, v. 1, n. 1, p. 21–29, 2014. Citado na página 21.

FERGUSON, A. G. The rise of big data policing: Surveillance, race, and the future of law enforcement. In: **The Rise of Big Data Policing**. [S.l.]: New York University Press, 2017. Citado na página 24.

GENDER API. 2024. Accessed: May 23, 2024. Disponível em: <https://gender-api.com/en>. Citado na página 48.

Governo do Estado de São Paulo. **Central de Atendimento**. 2024. Acessado: 21 de maio de 2024. Disponível em: <http://www.educacao.sp.gov.br/central-de-atendimento/index_escolas.asp>. Citado na página 41.

GUNAWAN, R.; RAHMATULLOH, A.; DARMAWAN, I.; FIRDAUS, F. Comparison of web scraping techniques: regular expression, html dom and xpath. In: ATLANTIS PRESS. **2018 International Conference on Industrial Enterprise and System Engineering (ICoIESE 2018)**. [S.l.], 2019. p. 283–287. Citado na página 53.

GUYON, I.; ELISSEEFF, A. An introduction to variable and feature selection. **Journal of machine learning research**, v. 3, n. Mar, p. 1157–1182, 2003. Citado na página 30.

HAIR, J. F.; BLACK, W. C.; BABIN, B. J.; ANDERSON, R. E.; TATHAM, R. L. **Análise multivariada de dados**. [S.l.]: Bookman editora, 2009. Citado na página 24.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. H.; FRIEDMAN, J. H. **The elements of statistical learning: data mining, inference, and prediction**. [S.l.]: Springer, 2009. v. 2. Citado 2 vezes nas páginas 33 e 34.

HEINES, R.; DICK, C.; POHLE, C.; JUNG, R. The tokenization of everything: Towards a framework for understanding the potentials of tokenized assets. In: **PACIS**. [S.l.: s.n.], 2021. p. 40. Citado na página 31.

HUNTER, J. D. **Matplotlib: A 2D Graphics Environment**. [S.l.], 2007. <https://matplotlib.org/stable/users/index.html>. Citado na página 35.

JURISTO, N.; MORENO, A. M.; SANCHEZ-SEGURA, M.-I. Analysing the impact of usability on software design. **Journal of Systems and Software**, Elsevier, v. 80, n. 9, p. 1506–1516, 2007. Citado na página 49.

KOHAVI, R.; DENG, A.; FRASCA, B.; LONGBOTHAM, R.; WALKER, T.; XU, Y. Trustworthy online controlled experiments: Five puzzling outcomes explained. In: **Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining**. [S.l.: s.n.], 2012. p. 786–794. Citado na página 24.

LIASHCHYNSKYI, P.; LIASHCHYNSKYI, P. Grid search, random search, genetic algorithm: a big comparison for nas. **arXiv preprint arXiv:1912.06059**, 2019. Citado 2 vezes nas páginas 35 e 56.

MAHER, N. A.; SENDERS, J. T.; HULSBERGEN, A. F.; LAMBA, N.; PARKER, M.; ONNELA, J.-P.; BREDENOORD, A. L.; SMITH, T. R.; BROEKMAN, M. L. Passive data collection and use in healthcare: A systematic review of ethical issues. **International journal of medical informatics**, Elsevier, v. 129, p. 242–247, 2019. Citado na página 25.

MURPHY, K. P. **Machine learning: a probabilistic perspective**. [S.l.]: MIT press, 2012. Citado 2 vezes nas páginas 32 e 35.

PENMAN, R. B.; BALDWIN, T.; MARTINEZ, D. Web scraping made simple with sitescraper. **Penman Web Scraping**, Citeseer, p. 1–10, 2009. Citado na página 53.

PERSSON, E. **Evaluating tools and techniques for web scraping**. 2019. Citado na página 26.

PROBST, P.; BOULESTEIX, A.-L.; BISCHL, B. Tunability: Importance of hyperparameters of machine learning algorithms. **Journal of Machine Learning Research**, v. 20, n. 53, p. 1–32, 2019. Citado na página 35.

PROVOST, F.; FAWCETT, T. **Data Science for Business: What you need to know about data mining and data-analytic thinking**. [S.l.]: "O'Reilly Media, Inc.", 2013. Citado na página 23.

RADENKOVIC, D.; KEOGH, S. B.; MARUTHAPPU, M. Data science in modern evidence-based medicine. **Journal of the Royal Society of Medicine**, SAGE Publications Sage UK: London, England, v. 112, n. 12, p. 493–494, 2019. Citado na página 23.

RAO, G. A.; SRINIVAS, G.; RAO, K. V.; REDDY, P. P. A partial ratio and ratio based fuzzy-wuzzy procedure for characteristic mining of mathematical formulas from documents. **IJSC—ICTACT J Soft Comput**, v. 8, n. 4, p. 1728–1732, 2018. Citado na página 30.

RASCHKA, S. Python machine learning. Packt Publishing, 2015. Citado na página 32.

RUSSELL, S. J.; NORVIG, P. Artificial intelligence: a modern approach (international edition). {Pearson US Imports & PHIPEs}, 2002. Citado na página 32.

SCHÜTZE, H.; MANNING, C. D.; RAGHAVAN, P. **Introduction to information retrieval**. [S.l.]: Cambridge University Press Cambridge, 2008. v. 39. Citado 2 vezes nas páginas 31 e 35.

SCIKIT-LEARN: Machine Learning in Python. 2024. <https://scikit-learn.org/stable/>. Accessed: 19/05/2024. Citado 4 vezes nas páginas 13, 34, 55 e 57.

SNELL, J.; MENALDO, N. Web scraping in an era of big data 2.0. **Bloomberg Law News**, 2016. Citado na página 25.

STOET, G.; GEARY, D. C. The gender-equality paradox in science, technology, engineering, and mathematics education. **Psychological science**, Sage Publications Sage CA: Los Angeles, CA, v. 29, n. 4, p. 581–593, 2018. Citado na página 21.

TORRES, A. Technovation challenge: Introducing innovation and mobile app development to girls around the world. In: **Mobile Media Learning: Innovation and Inspiration**. [S.l.: s.n.], 2015. p. 171–195. Citado na página 21.

VIJAYARANI, S.; JANANI, R. *et al.* Text mining: open source tokenization tools-an analysis. **Advanced Computational Intelligence: An International Journal (ACII)**, v. 3, n. 1, p. 37–47, 2016. Citado na página 31.

VISA, S.; RAMSAY, B.; RALESCU, A. L.; KNAAP, E. V. D. Confusion matrix-based feature selection. **Maics**, v. 710, n. 1, p. 120–127, 2011. Citado na página 36.

WASKOM, M. L. Seaborn: statistical data visualization. **Journal of Open Source Software**, v. 6, n. 60, p. 3021, 2021. Citado na página 36.

WINSHIP, C.; MARE, R. D. Models for sample selection bias. **Annual review of sociology**, Annual Reviews 4139 El Camino Way, PO Box 10139, Palo Alto, CA 94303-0139, USA, v. 18, n. 1, p. 327–350, 1992. Citado na página 52.

ZEBARI, R.; ABDULAZEEZ, A.; ZEEBAREE, D.; ZEBARI, D.; SAEED, J. A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. **Journal of Applied Science and Technology Trends**, v. 1, n. 1, p. 56–70, 2020. Citado na página 30.

APÊNDICE A

CÓDIGOS IMPLEMENTADOS

```
from fuzzywuzzy import fuzz, process
from unidecode import unidecode

def find_real_school(df, state_schools_df, threshold):
    df_copy = df.copy()
    df_copy["normalized_school"] = ""
    correspondency_dict = {}
    all_schools = state_schools_df["School_name"].str.upper().to_list()

    for i, row in df_copy.iterrows():
        if row["City"] is not None:
            city_name = unidecode(row["City"]).upper()

            # Create a list with schools from the students' city
            schools_from_city = state_schools_df[state_schools_df['City'].str.upper() == city_name]
            city_schools = schools_from_city["School_name"].str.upper().to_list()

        else:
            city_name = None
            city_schools = []

        school_name = row["School_name"].upper()

        if school_name in correspondency_dict:
            df_copy.at[i, "normalized_school"] = correspondency_dict.get(school_name)
            continue

        # Filter schools by city and attempt match
        matched_school_name = match_school(school_name, city_schools, all_schools, threshold)

        if matched_school_name:
            df_copy.at[i, "normalized_school"] = matched_school_name
            correspondency_dict[school_name] = matched_school_name
        else:
```

```
            correspondency_dict[school_name] = school_name

    return df_copy

def match_school(school_name, city_schools, all_schools, threshold):
    # Attempt match within city schools
    if city_schools != []:
        matches = process.extractOne(school_name, city_schools, scorer=fuzz.UQRatio)
        if matches[1] > threshold:
            return matches[0]

    # Fall back to matching within all schools
    matches = process.extractOne(school_name, all_schools, scorer=fuzz.UQRatio)
    if matches[1] > threshold:
        return matches[0]
    return None
```

Código-fonte 1: schoolfinder

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import GridSearchCV, train_test_split
from sklearn.metrics import make_scorer, f1_score, classification_report, confusion_matrix
import seaborn as sns

def evaluate_model(model_ref, param_grid, df):
  X = df.drop("target", axis=1)
  y = df["target"]
  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

  model = model_ref()
  custom_scorer = make_scorer(f1_score, pos_label=1)

  grid_search = GridSearchCV(estimator=model, param_grid=param_grid, cv=3, scoring=custom_scorer, verbose=1)

  sample_weights = np.ones_like(y_train)
  sample_weights[y_train == 1] = 5

  grid_search.fit(X_train, y_train)

  print("Best parameters:", grid_search.best_params_)
  print("Best score:", grid_search.best_score_)

```

```
25  best_model = grid_search.best_estimator_
26  predictions = best_model.predict(X_test)
27
28  print(classification_report(y_test, predictions))
29
30  conf_matrix = confusion_matrix(y_test, predictions)
31  plt.figure(figsize=(4, 3))
32  sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues", xticklabels
     =['0', '1'], yticklabels=['0', '1'])
33  plt.show()
34
35  probabilities = best_model.predict_proba(X_test)[:, 1]
36  threshold = 0.55
37  predictions = np.where(probabilities > threshold, 1, 0)
38  conf_matrix = confusion_matrix(y_test, predictions)
39  plt.figure(figsize=(4, 3))
40  sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues", xticklabels
     =['0', '1'], yticklabels=['0', '1'])
41  plt.show()
42  return best_model
```

Código-fonte 2: grid_search algorythm

```
1 import pandas as pd
2 from sklearn.linear_model import SGDClassifier
3 from sklearn.feature_extraction.text import CountVectorizer
4 from sklearn.pipeline import Pipeline
5 from sklearn.model_selection import train_test_split
6 import numpy as np
7
8 def graduation_classifier(df_info: pd.DataFrame, candidates_info: pd.
    DataFrame):
9     X = df_info["Graduation"]
10    y = df_info["Target"]
11
12    X_train, X_test, y_train, y_test = train_test_split(X, y, shuffle=True,
     test_size=0.15)
13    text_clf_sgd = Pipeline([('vect', CountVectorizer(stop_words=["como", "
    universidade", "MBA", "Doutorado", "Mestrado", "Sou", "de", "em", "USP",
     "ufscar"])), ('tfidf', TfidfTransformer()),
14                             ('clf-sgd', SGDClassifier(loss='hinge', penalty
    ='l2',alpha=1e-3, max_iter=1000, random_state=42))])
15
16    text_clf_sgd = text_clf_sgd.fit(X_train, y_train)
17    predicted_sgd = text_clf_sgd.predict(X_test)
18    np.mean(predicted_sgd == y_test)
19    predicted_sgd = text_clf_sgd.predict(candidates_info["Graduation"])
20
```

```
21     return predicted_sgd
```

Código-fonte 3: natural language classifier