ALINE LORENA TSURUDA
CAMILA MIWA IVANO
VINÍCIUS CARDIERI LOPEZ

AUTOMATIZAÇÃO DE ANÁLISE DE EMPRESAS PARA AUXÍLIO DE DECISÃO DE INVESTIMENTOS

São Paulo
2022

ALINE LORENA TSURUDA
CAMILA MIWA IVANO
VINÍCIUS CARDIERI LOPEZ

AUTOMATIZAÇÃO DE ANÁLISE DE EMPRESAS PARA AUXÍLIO DE DECISÃO DE INVESTIMENTOS

Trabalho apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Bacharel em Engenharia Elétrica - Ênfase Computação.

São Paulo
2022

ALINE LORENA TSURUDA
CAMILA MIWA IVANO
VINÍCIUS CARDIERI LOPEZ

AUTOMATIZAÇÃO DE ANÁLISE DE EMPRESAS PARA AUXÍLIO DE DECISÃO DE INVESTIMENTOS

Trabalho apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Bacharel em Engenharia Elétrica - Ênfase Computação.

Orientador:

Prof. Dr. Reginaldo Arakaki

Co-orientador:

Me. Victor Takashi Hayashi

São Paulo
2022

BANCA EXAMINADORA:

AGRADECIMENTOS

Aos meus pais Rinaldo e Rosangela que me apoiaram durante todo o período de faculdade. Ao meu irmão que me ouviu em todas as dificuldades. A todos os professores e orientadores que tive durante o período da Graduação e possibilitaram que eu chegasse a esse momento e meus colegas Camila e Vinícius, sem os quais nada disso seria possível.

Aline

A Deus, por estar aqui com saúde. Aos meus pais, Claudia e Wilson, por possibilitarem meus estudos, sempre me apoiando e acreditando em mim. À minha irmã, Emi, por torcer e celebrar minhas conquistas. Ao meu namorado, Rafael, por me acalmar em momentos difíceis. Aos companheiros dessa aventura, Vinícius e Aline, por ajudarem na realização desse sonho. Aos professores por nos guiarem nesse processo. E a todos que aqueles que contribuíram, de alguma forma, para a realização deste trabalho.

Camila

Aos meus pais, Jesus e Miriam, minha irmã, Bianca, meus amigos, colegas e namorada Marina, pelo apoio, incentivo e confiança na minha capacidade ao longo desta caminhada. Aos professores e à instituição pela honra de participar e me motivar a querer contribuir com uma sociedade melhor.

Vinicius

RESUMO

A decisão de seguir em frente em um investimento ou não tem se tornado cada vez mais estratégica e importante tanto no ciclo de vida das empresas, através dos fundos e rodadas de investimentos, em especial para empresas nascentes/*startups*, quanto para projetos de pesquisa ou desenvolvimento, através de agências de fomento ou editais de incentivo, como por exemplo FAPESP, FINEP e Fundo Patrimonial Amigos da Poli. Dentro desse tema, existem diferentes estruturas para analisar e viabilizar o investimento, a depender do tipo de projeto candidato, da fase de maturidade e outros fatores.

A partir desse cenário, o objetivo deste trabalho é criar uma metodologia de análise para auxiliar a decisão de investimento através de estratégias de automação da análise de dados e algoritmos de aprendizado de máquina de classificação, aplicado ao recorte de startups. A tese de um determinado investidor (usuário) é entendida implicitamente na modelagem a partir de um histórico de decisões prévias e serve de treino para a avaliação de uma nova empresa candidata. Desse modo, é automatizada uma parte do processo de avaliação e triagem, atualmente manual através de formulários de inscrição muitas vezes sem feedbacks para os empreendedores.

Palavras-Chave – Tomada de decisão, investimento, startup, análise de dados automatizada, algoritmo *k-nearest neighbors*, algoritmo *random forest*, análise automática, aderência à tese, *feedback* de análise.

ABSTRACT

The decision to commit to an investment rather than not has been a vital and strategic step in the development of both new enterprises, through investment funds and series funding especially for start-ups, and research projects, through development agencies and universities endowmnents. This theme has a broad set of structures and agents to analyze different opportunities based on the scope of the candidate project, its purpose, maturity and other factors.

This work aims to develop a methodology to conduct an automated data analysis and finally support the decision regarding an investment by implementing machine learning classification methods, focusing on the startups scenario. The investment thesis is implied in the modeling from a historic database with previous decisions as it provides a training database to evaluate new candidates. Therefore, this system automate part of the evaluation and sorting process, currently highly manual and dependant on subscription forms.

Keywords – Decision-Making, investment, startup, k-nearest neighbors algorithm, random forest algorithm, automatic analysis, adherence to thesis, feedback.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

PARTE I

INTRODUÇÃO

O capital de risco e investimento em *startups*, com o objetivo de alavancar seu crescimento, vem se destacando como fonte de financiamento, principlamente para a inovação(5) e a cada ano que passa, alcança números recordes no Brasil e no mundo. Só em 2021, esse mercado movimentou mais de USD 9bi segundo levantamento feito pela plataforma Distrito (6). Esse valor foi o maior registrado desde o início do estudo, em 2011 e é 2,5 vezes maior que o investido em 2020. Analisando globalmente, apesar de significativa queda em relação ao mesmo período do ano passado, no terceiro quadrimestre de 2022, foram investidos cerca de USD74,5bi, segundo *report* realizado pelo CBInsights (7).

Apesar de um mercado novo, sobretudo no Brasil, os Fundos de Investimento em *startups*, conhecidos como *Private Equity* - investimentos em empresas não listadas na Bolsa, mas já consolidadas e relevantes no mercado de atuação - e *Venture Capital* - investimento em novas *startups* e empresas emergentes, sendo movimentações de alto risco - também aumentam de número a cada ano, com mais empresas dedicadas ao negócio, mas também com empresas que começaram a investir em *startups* de acordo com seus setores, criando os *Corporate Venture Capital*. O objetivo desses fundos é investir em empresas em estágios iniciais, mas que apresentam alto potencial, atrelado, no entanto, a um alto risco (8).

Segundo a Forbes (9), independente do tamanho do fundo, corporativo ou não, os principais critérios utilizados para esses investimentos são, entre outros:

- Fase de crescimento da empresa: empresas com receita e fluxo de caixa sólidos apresentam menor risco. No entanto, *startups* recentes, apesar do maior risco, apresentam maior potencial de crescimento no curto prazo;
- Tipos de empresas: o setor da empresa, se ela é baseada em tecnologia ou não, seu foco e segmento são importantes na construção de teses;
- Tamanho do Investimento: o tamanho da captação pretendida pela empresa, ou seja, a necessidade de capital para atingimento das metas pode ser um impeditivo para alguns fundos;
- Sócios: os investimentos em *startups* são de longo prazo, normalmente por no mínimo 5 anos, o tempo dos fundos e os gestores precisam confiar que as lideranças da empresa são as adequadas para escalar e fazer o negócio crescer

No entanto, são justamente nesses critérios que os fundos se diferenciam entre si. Existem fundos que investem em setores específicos, como saúde, agronegócio, logística,

etc., outros se diferenciam por olhar empresas que já possuem clientes e receita recorrente, diminuindo assim seu fator de risco. Alguns fundos não investem apenas no empreendedor fundador, e gostam de que já haja uma equipe multidisciplinar formada.

A junção desses critérios é denominada tese de investimento e suas características macro podem ser encontradas, normalmente, no *site* do próprio fundo. Essa é a visão que os fundos conseguem passar para o empreendedor que busca uma rodada de investimento e que procura o Fundo em que a empresa dele possua maior chance de ter um investimento aprovado e que mais possa ajudá-lo a crescer.

1 OBJETIVO DO PROJETO

O objetivo deste estudo foi desenvolver uma ferramenta semiautomática e inteligente que auxilia na avaliação de empresas de capital privado que estão em busca de aceleração ou investimento. Tal ferramenta é capaz de retornar ao usuário informações relevantes sobre a empresa analisada que auxiliam na tomada de decisão de investimentos. As características estudadas são possíveis pontos de melhoria, como o nível de maturidade da empresa e retorna uma resposta binária indicando ou não probabilidade de investimento de um investidor específico em uma empresa.

Essa análise é feita a partir de um banco de dados referente às decisões anteriores de cada perfil de investidor, isto é, empresas previamente analisadas cujas decisões de investimento já foram tomadas de acordo com a tese de investimento do usuário-investidor. O objetivo é que o investidor cadastre seu banco de dados, com os parâmetros analisados das empresas que já passaram pelo Fundo, tendo sido ou não apoiadas. Em seguida o investidor pode cadastrar empresas que terão uma avaliação com base nesse histórico. O fluxo para o outro usuário (usuário empreendedor) envolve um questionário sobre sua empresa e a escolha de uma tese já cadastrada na plataforma para que o sistema possa analisá-la e retornar um relatório.

A ferramenta serve como mais um parâmetro a ser avaliado pelo investidor na tomada de decisão e análise de uma empresa, sendo importante que o investidor ainda mantenha sua análise padrão qualitativa e de perfil dos empreendedores.

2 METODOLOGIA E IDEAÇÃO

2.1 Metodologia do projeto

Para o desenvolvimento, o grupo se guiou pelo seguinte método:

1. Insights da disciplina de empreendedorismo (PCS-3529 - Criação e Administração de Empresas de Computação (10))
2. Estudo de contexto e pesquisa detalhada
3. Entrevista com potenciais usuários
4. Reuniões quinzenais de acompanhamento para desenvolvimento
5. Exercício de testes de instâncias conhecidas

2.2 Definição do assunto

A disciplina de empreendedorismo foi o principal ponto de partida para a identificação das motivações e proposta de trabalho, uma vez que foi a oportunidade de se analisar minuciosamente a jornada do empreendedor que será investido, além de ter um contato com um gestor de um fundo de *Venture Capital* que comentou sobre aspectos analisados por ele e pelo time para considerar ou não um investimento.

Após o término da disciplina, o grupo conduziu por conta própria uma pesquisa mais aprofundada para ter uma noção melhor definida do contexto em que os empreendedores estão inseridos e o que está sendo analisado nos momentos de decisão de investimento. Dentre as referências buscadas, o grupo focou primeiro naquelas voltadas aos primeiros passos aconselhados para os empreendedores, a fim de entender a jornada pela qual eles passarão. Em seguida, analisou uma base de dados disponibilizada publicamente e referência no mercado de mapeamento de mercado para ter exemplos reais e passou à etapa

de construção de ontologias para definir uma relação entre os dados analisados com o conceito de empresa e traduzir esses exemplos reais de acordo com a trilha de desenvolvimento proposta na literatura.

2.3 Prototipação para validação da nossa proposta

O passo seguinte refere-se a uma validação das ideias mapeadas e das propostas feitas pelo grupo. Essas propostas envolvem a própria motivação do trabalho (oportunidades identificadas no ciclo de investimento), a estrturação dos dados, os algoritmos a serem desenvolvidos e a experiência de usuários.

Para isso, o grupo buscou um orientador que além do contato acadêmico com Inteligência Artifical e *Machine Learning* possuísse também contato com o Mercado Financeiro. Assim, além do apoio do orientador, com experiência no Setor Bancário, foi muito importante o apoio do co-orientador, com passagem por um Fundo de Investimentos para que o grupo validasse a aderência da Solução no Mercado.

Adicionalmente ao apoio do Orientador e Co-Orientador, o grupo realizou entrevistas com potenciais usuários do sistema, gestores do *family office*, Anima Investimentos, para aprofundar o conhecimento sobre como são feitas as análises hoje e principais dificuldades que os mesmos possuem (vide apêndice B). Após concluir os trabalhos, o grupo se reuniu novamente com esses gestores para obter sugestões e *feedbacks* dos mesmos (apêndice C).

2.4 Desenvolvimento do sistema

Na mesma direção de validação e ajustes, uma vez iniciado o desenvolvimento do sistema, o grupo entende como essencial o exercício de testes-modelos para conferência dos resultados obtidos. Para isso, durante todo o ano, o grupo se reuniu quinzenalmente com o Orientador e, nas quinzenas complementares, em reuniões de trabalho entre os próprios membros, para fazer *sprints* quinzenais de desenvolvimento e validação das etapas.

O método utilizado foi separar os membros em frentes específicas, que trabalharam paralelamente no desenvolvimento durante duas semanas e focavam na integração entre as partes do sistema quando reunidos para desenvolverem juntos. Foi estabelecido também em conjunto as linguagens prioritárias a serem usadas (Python) e o banco de dados utilizado no projeto (MongoDB).

Para o desenvolvimento dos algoritmos de *Machine Learning* e implementação da

rotina de análise de dados, o grupo se baseou na metodologia do *Data Science Pipeline* (11), figura 1. Desse modo, fora conduzida uma implementação incremental de cada passo destacado abaixo, conforme especificado na seção de desenvolvimento deste documento.

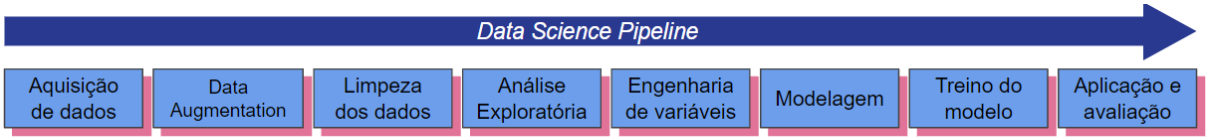


Figura 1: *Pipeline* em alto nível da frente de desenvolvimento do algoritmo de *Machine Learning* do projeto. Fonte: produzida pelos autores.

2.5 Validação e testes: Base de Dados, Sintetizando os dados, avaliando precisão

Para a parte de validação dos conceitos teóricos para o modelo em si, o grupo enfrentou dificuldade em obter os dados para compor essa base para treinamento do algoritmo. Do lado dos usuários-investidor, os investidores não mencionam detalhes de suas teses de investimento ou seus controles internos de por que aprovaram ou não uma empresa, apenas empresas investidas, *Cases* de sucesso. Porém mesmo para as empresas investidas, não se tem muitos dados, uma vez que são empresas de capital fechado, que precisam fazer relatórios e prestações de contas apenas para seus sócios, e não para o mercado, não sendo possível obter números exatos ou uma base estruturada para esses casos. E do outro lado, da análise de empresas para uma dada tese de investimentos, existem alguns repositórios com dados públicos sobre muitas empresas, como Crunchbase (12) e datasets disponibilizados pela plataforma Kaggle (13). Essas plataformas foram avaliados pelo grupo para se tornarem fonte de dados para o treinamento, porém boa parte das informações procuradas devem ser tratadas com confidencialidade, e portanto essas bases não se mostraram suficientes.

Dessa forma, o grupo decidiu seguir pela alternativa de sintetização de dados estruturados. Ainda assim esta possibilidade apresentou dificuldade, uma vez que há pouca literatura sobre dados estruturados, e muito conteúdo sobre dados não estruturados, como imagens. A solução do grupo foi elencar os pontos de mais fácil controle interno e criar uma base sintética com informações preenchidas randomicamente de maneira automática, somente a informação de investimento foi manualmente preenchida. A base sintética foi então avaliada, na sua precisão, significância e outras métricas relevantes.

PARTE II

EMBASAMENTO TEÓRICO

3 EMBASAMENTO TEÓRICO

O embasamento teórico será dividido em 2 grandes partes: Ciclo de vida de uma *startup*, em que será abordado a parte mercadológica e análises sobre empresas, sobretudo no Brasil, as diversas etapas que ela passa, e respectivas necessidades. A segunda parte está relacionada ao Aprendizado de Máquina e os modelos escolhidos e aplicados pelo grupo.

3.1 Ciclo de vida em uma *startup*

Tendo em vista a disciplina de Empreedorismo (10), o grupo guiou o trabalho entendendo o ciclo de vida de uma *startup* e os desafios que ela tem para se desenvolver. Isso se dá, principalmente por conta das pessoas do Mercado que acreditam na ideia. Segundo Geoffrey Moore (1), no começo de uma ideia, apenas os visionários acreditam nela. Trazendo para meios financeiros e a criação de empresas, isso representa o capital do próprio empreendedor, e, em alguns casos, de família e amigos e alguns poucos representantes, chamados de "primeiros adotantes". Dos primeiros adontates até chegar na Maioria Inicial, há um grande abismo, que é onde a maior parte das empresas acaba falindo. Esse é o momento que elas buscam os fundos de *Venture Capital*. Depois da Maioria Inicial, uma vez consolidado o produto e tecnologia, há a adoção pela Maioria Tardia, representando os Fundos de *Private Equity* e por fim os retardatários.

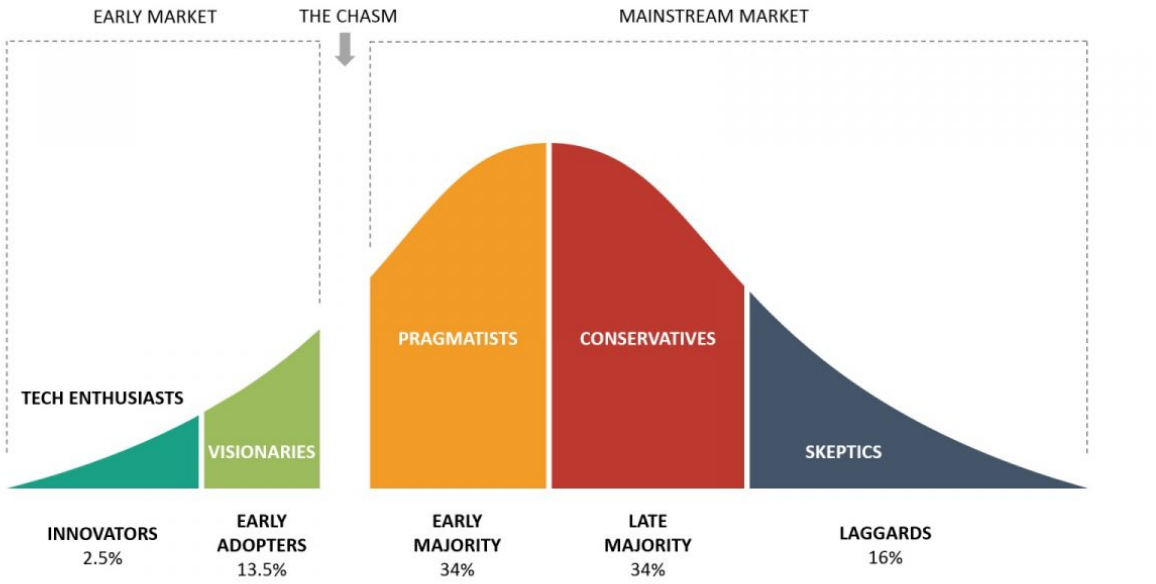


Figura 2: Grupos de adoção da tecnologia e o abismo. Fonte: Crossing the Chasm (1).

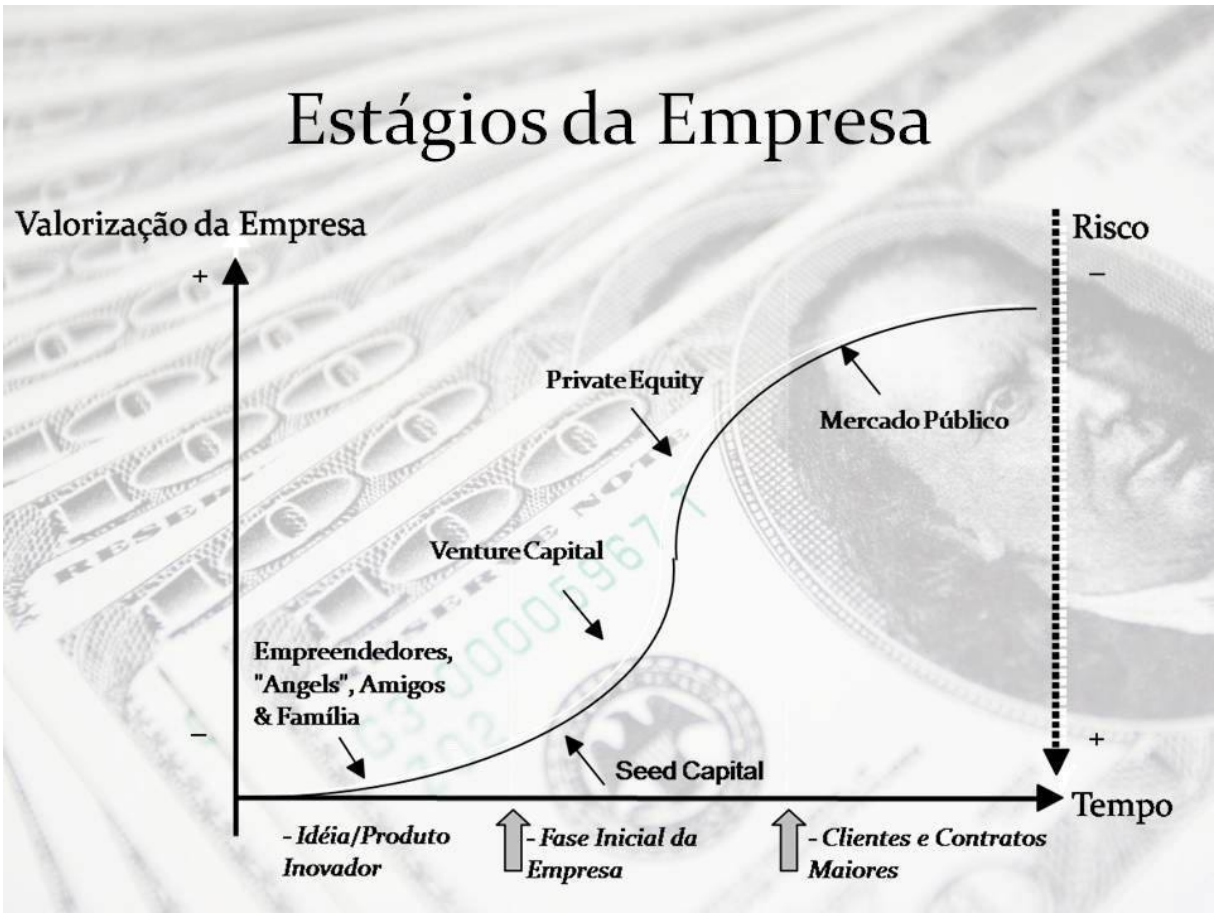


Figura 3: Estágios de uma empresa de acordo com o tempo e risco. Fonte: Endeavor (2).

Esse funil e abismo, podem ser observado na prática, uma vez que não são todas as *startups*, mesmo recebendo investimentos, que conseguem crescer e superar os desafios. Observando o Ciclo de Investimentos, segundo a Distrito(3) apenas 0,8% das empresas que recebem algum investimento inicial conseguem chegar a um investimento série E, ou seja, ter passado por 5 ou mais rodadas anteriores.

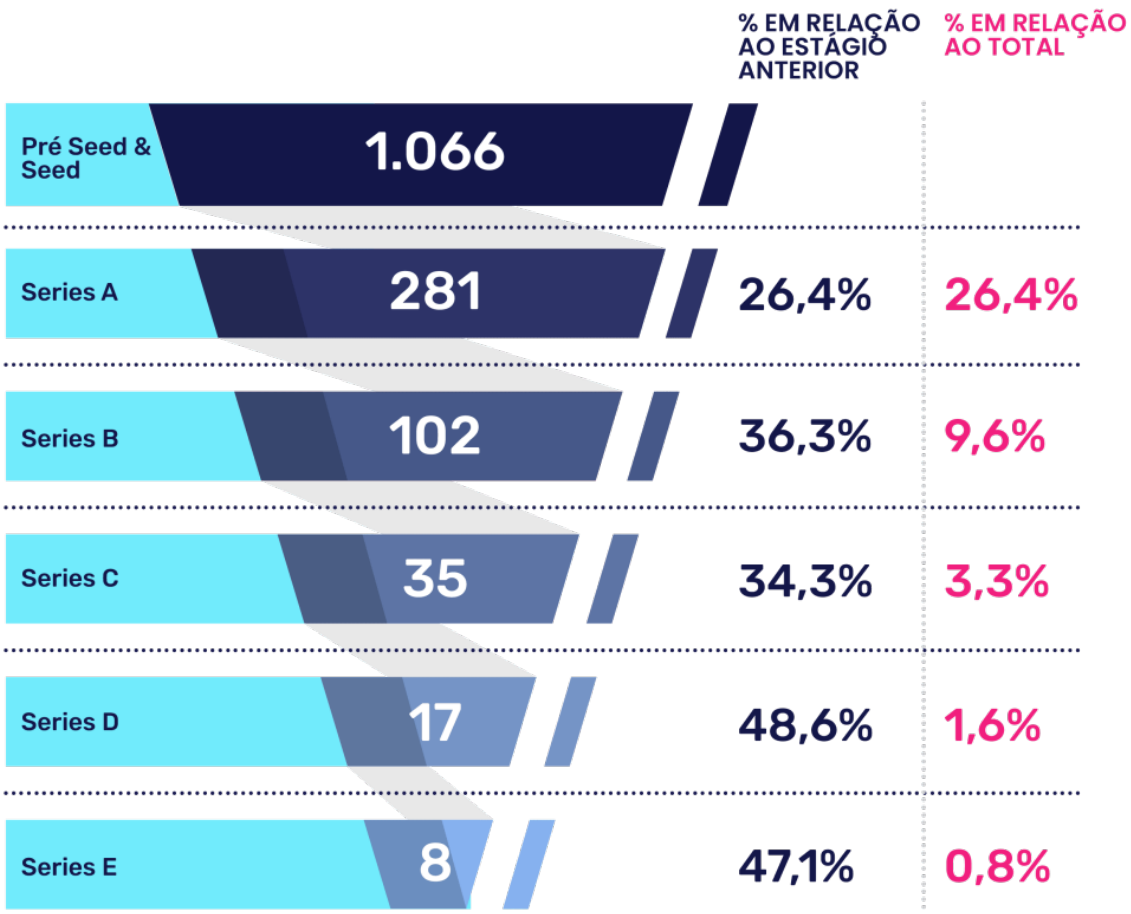


Figura 4: Funil de conclusão de rodadas de Investimento. Fonte: Distrito (3).

Considerando então a necessidade de conquista da Maioria Inicial (1), o primeiro passo foi a definição dos critérios mais comumente utilizados para avaliação de empresas. Como base teórica para definição destas características, partiu-se dos 5 estágios das empresas definidos por Churchill e Lewis (14): existência, sobrevivência, sucesso, decolagem e maturidade. Para cada estágios, os autores determinam algumas características básicas que as empresas devem apresentar. São elas:

- Estilo da liderança: enquanto no estágio de existência, a liderança deve apresentar supervisão direta. Conforme o negócio for evoluindo, essa supervisão diminui e a responsabilidade é passada para os funcionários.

- Organização: no início da empresa, a organização é basicamente dos sócios fundadores. Com o crescimento da mesma, é necessária uma organização com várias equipes e funcionários para suportarem as novas demandas e mudanças

- Existência de sistemas formais: esses sistemas acabam não sendo prioridade na constituição de uma empresa, porém conforme seu quadro e seus negócios forem crescendo, a fim de manter a organização e ter uma governança estabelecida, é necessário haver sistemas formais em toda a empresa

- Relação entre os sócios e a empresa: no início da existência de um negócio, ele é muito dependentente dos sócios fundadores, mesmo que os mesmos não dediquem todo o seu tempo a ela. Conforme o negócio cresce, os sócios passam a dedicar todo o seu tempo à empresa, mas a mesma deve ser capaz de ter sua operação independente dos fundadores.

Essas características podem ser segmentadas em outras mais específicas, com métricas quantificáveis, como anos de operação da empresa, número de sócios que se dedicam integralmente ao negócio, número de funcionários, receita da empresa, entre outros, que serão explorados mais à frente.

3.2 Aprendizado de máquina

Conforme estudado na disciplina de Inteligência Artificial (PCS3438) (15) ministrada pela professora Anna Helena Reali Costa e pelo professor Eduardo Raul Hruschka, no curso de graduação de Engenharia Elétrica e ênfase de Computação da Escola Politécnica, com a evolução da inteligência artificial, está cada vez mais comum o uso de algoritmos de Aprendizado de Máquina para resolver problemas complexos, uma vez que com eles, é possível que as máquinas aprendam, sem necessidade de uma programação específica para uma ação, e sim apenas do algoritmo. Esse algoritmo diferencia-se da programação usual, visto que tradicionalmente era necessário desenvolver sistemas explicitando passo a passo no código, ou seja, esses algoritmos podem solucionar os problemas de forma autônoma, por isso conseguem analisar dados de entrada e prever possíveis saídas, treinando seus modelos.

Existem diferentes métodos de classificação em *Machine Learning*, os escolhidos neste trabalho são os algoritmos *K-Nearest Neighbours* (K-NN) e *Random Forest*. O K-NN classifica um novo registro com base na proximidade com outros K registros mais próximos

e suas classificações. O modelo pode ser parametrizável, definindo por exemplo o valor K e o método de cálculo de distância entre os registros. O *Random Forest* é um modelo mais avançado baseando em Árvores de Classificação, basicamente este modelo simula mais de uma árvore de decisão com diferentes bases de treinamento. Este modelo também pode ser configurável, a otimização pode ser feita com base no número de árvores geradas e na profundidade da árvore, por exemplo.

Para definir um modelo, o algoritmo analisa uma base de dados fornecida e o divide em três partes: treinamento, validação e testes. Os dados de treinamento são usados para treinar o modelo desejado, e os de validação servem para comparar diferentes modelos e hiperparâmetros. Esses dois conjuntos de dados são usados na etapa de treinamento para definir o modelo a ser utilizado. Nesse processo do treinamento, uma das técnicas utilizadas é o *cross-validation*, na qual a base é divida entre treino e validação de diversas maneiras, para evitar problemas de *overfitting*. E por fim, os dados de teste são utilizados para comprovar que o modelo obtido atua de maneira esperada. A partir desses resultados, parte-se para a fase de análise dos modelos.

Além do treino do modelo em si, a análise dos modelos e dos resultados obtidos são igualmente importantes. Neste projeto, duas métricas tiveram alta relevância, acurácia e *recall*. A acurácia indica a taxa geral de classificações corretas, dando uma avaliação macro do modelo, e o *recall* reporta a taxa de classificação correta dos positivos. A acurácia foi relevante em dois momentos, na reestruturação da quantidade de parâmetros relevantes nos treinos, visando evitar valores extremamente altos, pois podem indicar *overfitting*; e na escolha do modelo para a análise de uma empresa, dado que quanto mais alta a acurácia, maior a probabilidade da análise automatizada estar correta. E a taxa de *recall* abaixo do esperado incentivou o uso de técnicas de *oversampling* para equilibrar os elementos negativos e positivos.

PARTE III

SISTEMA DE AUXÍLIO À AVALIAÇÃO E DECISÃO DE INVESTIMENTOS

4 DESCRIÇÃO DO SISTEMA

Entende-se como pré-requisitos a definição do processo de desenvolvimento, de técnicas e outros procedimentos necessários para o avanço do projeto. A proposta é a criação de um sistema que, apesar de ser complementar às soluções que já são utilizadas pelos fundos de investimento (de qualquer escala: de "Anjo" a "Venture Capital"), seja uma solução autônoma e útil para mitigar as dores dos usuários. O grupo considera a seguinte estrutura preliminar para o projeto:

- Mapeamento dos processos atuais dos investidores e das ferramentas utilizadas;
- Levantamento das características avaliadas num processo de análise de investimentos;
- Estruturação parametrizada dos dados a serem analisados;
  - Levantamento da literatura atual do tema de Empreendedorismo;
  - Comparação com o sistema proposto neste projeto;
- Implementação;
  - Criação do Banco de Dados e desenvolvimento de serviço de Interface de Programação de Aplicação (API) para comunicação com o *database*;
  - Porjetar um algoritmo de classificação baseado nos modelos K-NN e *Random Forest*;
  - Integrar partes e fornecer ao usuário através de uma página *web*;

A proposta dos dois primeiros pré-requisitos é compreender e posicionar o projeto de acordo com as necessidades do mercado e assim garantir que o mesmo tenha valor enquanto uma solução útil. Assim, se faz necessário um entendimento amplo do processo existente atualmente e quais são suas bases, uma vez que a proposta de sistema desenvolvido é complementar e automatizar o processo atual, e não substituí-lo por completo.

Em um terceiro momento, o grupo estruturou os dados coletados a partir das etapas anteriores de maneira que reflita o processo atual mas que permita a adaptação às diferentes teses de investimento. A avaliação realizada pela plataforma se baseará em um banco de dados disponível, sabendo que os investidores naturalmente terão acesso ao seu próprio portfólio e histórico. Para aplicação no presente trabalho, o investidor precisa adaptar sua base histórica de investimento ao modelo fornecido, para que o sistema esteja compatível e consiga fazer a análise corretamente. Para implementação no mercado, a prioridade do grupo seria conseguir esboçar uma organização parametrizada dos dados, que realize a adaptação do banco de dados de cada investidor à ontologia desenhada pelo grupo, de modo que permita uma maleabilidade do sistema para as diferentes aplicações identificadas na motivação do trabalho. A ontologia é uma maneira de estruturar um conceito (no caso discutido foi importante definir o conceito "empresa"), relacionando características sobre o conceito central. Dessa maneira, a ontologia permite que os dados se relacionem e a o processo de adaptação de diferentes fontes de informação pode ser automatizada com mais facilidade. A ontologia estrturada se encontra detalhada mais adiante na figura 7.

Entende-se como principal requisito a estrutura personalizável dos dados, já que neste tema - análises de desempenho econômico - é notório o comportamento dinâmico dos resultados e a importância de seus parâmetros. Devido a esse desempenho tão pouco preditivo, o grupo entende como fundamental para a relevância do sistema, a aderência ao conceito de *"explainable ai"* - não se limitando a uma resposta fechada oriunda de qualquer algoritmo de classificação por IA, mas sim fornecendo um relatório com as tendências observadas ao comparar a empresa candidata com o histórico de investimentos. Levando isso tudo em consideração, o grupo escolheu como algoritmo de classificação os modelos K-NN e *Random Forest*. Entende-se ainda que em contribuições futuras, se possa partir da ontologia e estrutura desenvolvidas neste trabalho para analisar o desempenho de algoritmos mais complexos.

4.1 Tipos de usuários

O projeto desenvolvido olha dois tipos de usuários, o investidor e o empreendedor. No sistema proposto, cada usuário investidor pode definir sua tese de decisão de investimento. Cada tese do investidor é modelada de maneira automatizada, com base no histórico de investimentos fornecido pelo investidor. Cada usuário empreendedor pode entrar com os dados da sua empresa, para gerar o a análise automatizada para uma tese já cadastrada

na plataforma.

O empreendedor é o usuário dono de uma empresa que procura por um aporte financeiro, seu objetivo com o projeto é receber uma avaliação sobre a maturidade da própria empresa com base no feedback dado, indicando quais pontos já estão maduros, e quais ainda podem ser melhorados. E o investidor é o usuário que procura por oportunidades de investimento, tendo em mente que o investimento em empresas pode retornar grandes rentabilidades no futuro.

Ao realizar uma análise, ambos os lados, dono da tese (investidor) e o empreendedor, recebem os comentários finais, indicando se a análise retornou um resultado favorável ao investimento, e em caso negativo, possíveis alterações na empresa que aumentem suas chances de capitalização de investimento.

4.2 Requisitos funcionais

Os requisitos funcionais do sistema foram levantados e estão listados na tabelas a seguir. Estes requisitos expressam as funcionalidades essenciais para que o sistema funcione conforme o esperado. A coluna ID relaciona cada descrição de funcionalidade com um identificador, para facilitar mencionar ao longo do texto. E a coluna usuário foi acrescentada para indicar o ator relacionado a tal tarefa, o usuário "(Interno)" indica que a tarefa é unicamente do sistema, sem precisar de ações de outros atores.

ID	Descrição	Usuário
RF1	Receber e armazenar os dados fornecidos sobre a empresa a ser analisada	Empreendedor
RF2	Receber os dados de investimento históricos fornecidos pelo investidor para gerar a tese de investimento	Investidor
RF3	Armazenar o histórico de investimento, para gerar os modelos que realizarão a análise das empresas	(Interno)
RF4	Realizar o tratamento dos dados da base histórica, incluindo sintetização e parametrização de dados	(Interno)
RF5	Armazenar os dados intermediários no tratamento da base histórica, para agilizar o processamento	(Interno)
RF6	Realizar análises comparativas entre os modelos de classificação e decidir o melhor modelo para determinada base histórica	(Interno)
RF7	Gerar relatório com feedback para os dois tipos de usuário	Empreendedor e investidor

Tabela 1: Requisitos funcionais do sistema. Fonte: produzida pelos autores.

Além desses requisitos funcionais, como passos futuros, para utilização real no mercado, poderíamos acrescentar outras funcionalidades como automatizar a adaptação de qualquer base histórica de investimentos para um modelo que possa ser entendido pelo sistema e realizar todas as análises e verificar qual a maior compatibilidade entre empresa analisada e as teses cadastradas.

4.3 Requisitos não Funcionais

Os requisitos não funcionais do sistema projetado estão indicados na tabela a seguir. Requisito Não Funcional está relacionado a “como” fazer suas funções, por isso linkamos com aspectos de segurança e desempenho.

ID	Descrição	
RNF1	Usabilidade	Preenchimento do formulário de forma não morosa e trabalhosa
RNF2	Confiabilidade	O sistema deve estar sempre disponível
RNF3	Desempenho	O sistema deve realizar atividades de baixa latência, como login e cadastro, em menos de 1 minuto. E a análise de empresas pode demorar até 5 minutos
RNF4	Portabilidade	Sistema somente disponível para web page, por enquanto
RNF5	Segurança	Sistema deve seguir as normas da Lei Geral de Proteção de Dados (LGPD), garantir a segurança dos dados fornecidos e explicitar os objetivos do armazenamento dos dados
RNF6	Integridade	Sistema deve realizar o controle de acesso, isto é, perfil investidor não acessa funções empreendedor e vice versa
RNF7	Modificabilidade	Sistema deve ser parametrizável
RNF8	Rastreabilidade	Sistema deve registrar todas as tomadas de decisões, e como afetou o novo algoritmo, para evitar futuros resultados indesejáveis

Tabela 2: Requisitos não funcionais do sistema. Fonte: produzida pelos autores.

4.4 Diagrama de caso de uso

O sistema discutido possui os casos de uso relacionados a seguir. O diagrama com a relação de atores e casos de uso é apresentado na sequência.

1. Cadastra usuário
2. Preenche formulário do investidor e envia base histórica de investimentos
3. Modela tese de investimento
4. Preenche formulário sobre a empresa
5. Analisa a empresa para um tese de investimentos
6. Acessa resultados da análise

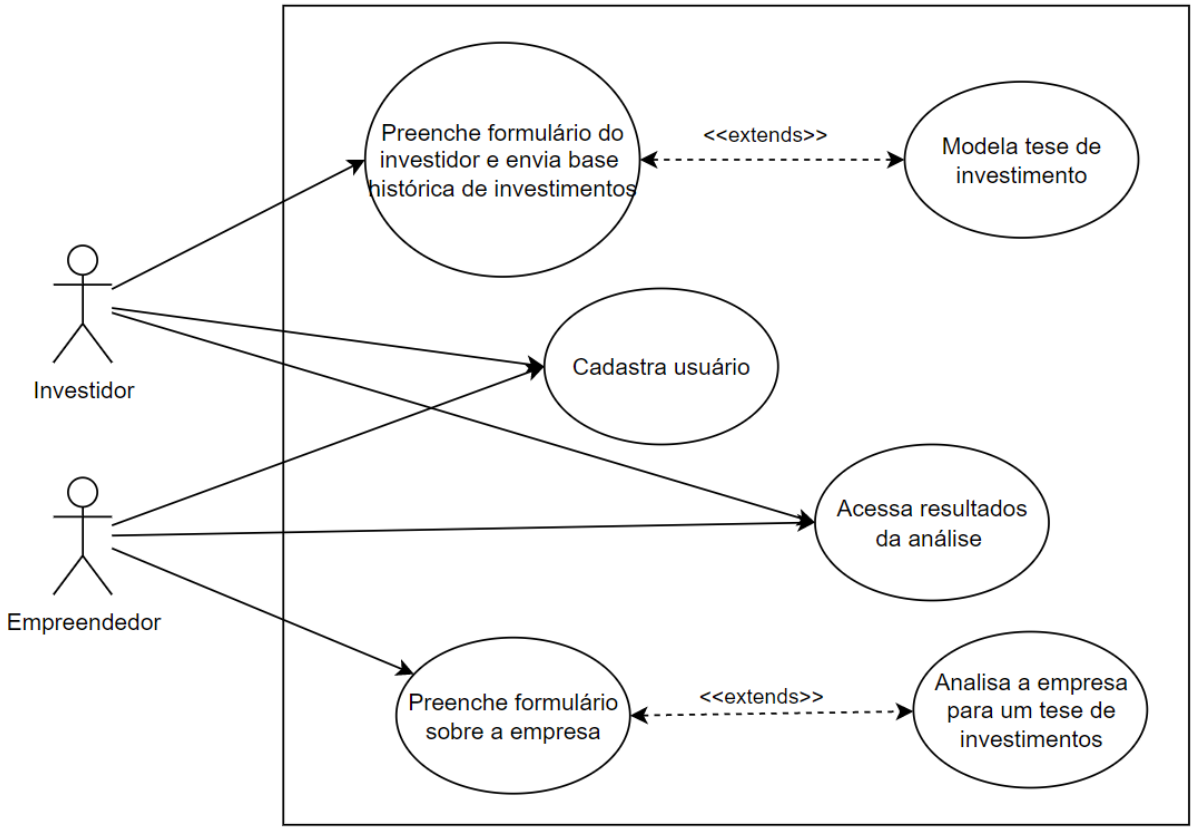


Figura 5: Diagrama dos casos de uso. Fonte: produzida pelos autores.

4.5 Detalhamento de cada processo:

4.5.1 Cadastra usuário

- Descrição: inicia o processo de cadastro de um usuário
- Ator: usuário (investidor ou empreendedor)
- Evento iniciador: o usuário acessa a plataforma, e preenche formulário para se cadastrar
- Pré-condição: o usuário acessou a plataforma
- Pós-condição: dados do usuário recebidos pela plataforma
- Fluxo Principal:
  - O usuário preenche seus próprios dados, incluindo novo usuário e senha

  - Em sequência, se o usuário se identificar como investidor, passa para o fluxo de envio de dados históricos, caso seja um usuário empreendedor, passa para o fluxo de envio de dados da empresa

4.5.2 Preenche formulário do investidor e envia base histórica de investimentos

- Descrição: envio dos dados históricos de investimento
- Ator: usuário-investidor
- Evento iniciador: o usuário acessa a plataforma, e se identifica como um investidor
- Pré-condição: o usuário acessou a plataforma como investidor
- Pós-condição: dados históricos de investimento recebidos pela plataforma
- Fluxo Principal:
  - O usuário define um nome para a tese a ser cadastrada
  - O usuário anexa a base histórica de investimento conforme o modelo fornecido pelo sistema
  - O usuário confirma que os dados enviados são verdadeiros, e que assume responsabilidade sobre a veracidade dos dados.
  - O usuário concorda com os nosso termos de consentimento (LGPD, armazenamento e uso de dados)
  - O usuário conclui o processo, enviando os dados, documentos e consentimento com os Termos
- Fluxo de exceção:
  - Usuário não concorda com os termos e/ou não confirma a veracidade dos dados: Sistema indica que é necessário concordar/consentir para prosseguir
  - Usuário não preenche todas as informações necessárias: Sistema indica que faltam informações para enviar o cadastro, e obter análise

4.5.3 Modela tese de investimento

- Descrição: modela uma tese de investimento em cima da análise sobre a base histórica de investimentos

- Ator: usuário-investidor
- Evento iniciador: o usuário acessa a plataforma, se identifica como um investidor, anexa base histórica de investimentos e confirma cadastro
- Pré-condição: o usuário acessou a plataforma como investidor, anexou a base histórica de investimentos e confirmou cadastro
- Pós-condições: tese modelada para uma base histórica de investimentos
- Fluxo Principal:
  - O usuário enviou sua base histórica de investimentos e confirmou cadastro
  - O sistema chama algoritmo de modelagem da tese
  - O sistema retorna para o usuário informações relevantes sobre o processo de modelagem
  - O usuário pode testar sua tese para uma empresa, seguindo para o fluxo de preenchimento de formulário da empresa, pulando o passo de escolha da tese. Ao final o usuário receberá o resultado da análise e, se houver, sugestões de mudança que possam aumentar as chances de aprovar investimento
- Fluxo de execução
  - Usuário envia formulário fora do padrão solicitado: Sistema indica que as informações são inválidas e que é necessário corrigi-las para prosseguir
  - Usuário não preenche todas as informações necessárias: Sistema indica que faltam informações para enviar o cadastro, e obter análise

4.5.4 Preenche formulário sobre a empresa

- Descrição: preenchimento de um formulário sobre a empresa para enviar os dados da empresa para o sistema, realizando o cadastro
- Ator: usuário-empreendedor
- Evento iniciador: o usuário acessa a plataforma como empreendedor
- Pré-condição: o usuário acessou a plataforma como empreendedor
- Pós-condição: dados da empresa recebidos pela plataforma

- Fluxo Principal:
  - O usuário preenche os dados sobre a empresa
  - O usuário confirma que os dados enviados são verdadeiros, e que assume responsabilidade sobre a veracidade dos dados
  - O usuário concorda com os nosso termos de consentimento (LGPD, armazenamento e uso de dados)
  - O usuário conclui o processo, enviando os dados, documentos e consentimento com os Termos
- Fluxo de exceção:
  - Usuário não concorda com os termos e/ou não confirma a veracidade dos dados: Sistema indica que é necessário concordar/consentir para prosseguir
  - Usuário não preenche todas as informações necessárias: Sistema indica que faltam informações para enviar o cadastro, e obter análise

4.5.5 Analisa a empresa para um tese de investimentos

- Descrição: análise da empresa com base em uma tese de investimento, para indicar ou não sugestão de aporte na empresa
- Ator: usuário-empreendedor
- Evento iniciador: preenche dados sobre a empresa a ser analisada
- Pré-condição:
  - Se usuário é investidor: já cadastrou sua base histórica de investimentos
  - Se usuário é empreendedor: o usuário acessou a plataforma, se identificou como um empreendedor, preencheu dados sobre a empresa e confirmou cadastro. E o usuário investidor já cadastrou sua base histórica de investimentos
- Pós-condições: empresa analisada para uma determinada tese de investimento
- Fluxo Principal:
  - Usuário preencheu dados sobre a empresa e confirmou cadastro
  - Usuário (empreendedor) seleciona uma tese já cadastrada na base
  - Sistema chama algoritmo de análise da empresa

  - Segue para o fluxo de acesso aos resultados da análise

- Fluxo de execução

  - Usuário não preenche todas as informações necessárias: Sistema indica que faltam informações para enviar o cadastro, e obter análise

4.5.6 Acessa resultados da análise

- Descrição: o sistema realiza a análise dos dados fornecidos sobre a empresa com base na tese de investimento definida
- Ator: usuário (investidor ou empreendedor)
- Evento iniciador:

  - Se usuário é investidor: preencheu dados sobre uma empresa a ser avaliada após enviar base histórica de investimentos
  - Se usuário é empreendedor: selecionou uma tese após cadastrar empresa

- Pré-condição:

  - Se usuário é investidor: realizou o cadastro e o envio da base histórica de investimento, e preencheu dados sobre uma empresa a ser avaliada
  - Se usuário é empreendedor: realizou cadastro, preencheu dados sobre a empresa a ser avaliada e selecionou uma tese

- Pós-condição: sistema retorna para o usuário o resultado da análise e, se houver, sugestões de mudança que possam aumentar as chances de receber investimento
- Fluxo Principal:

  - Se usuário é investidor:
    * O usuário acessou a plataforma
    * O usuário realizou cadastro como usuário investidor
    * O usuário enviou base histórica de investimentos
    * O sistema modelou tese de investimento
    * O usuário preencheu dados sobre uma empresa para ser analisada
    * O sistema chamou algoritmo de análise da empresa

        * O sistema retorna para o usuário o resultado da análise e, se houver, sugestões de mudança que possam aumentar as chances de receber investimento

    - Se usuário é empreendedor:
        * O usuário acessou a plataforma
        * O usuário realizou cadastro como usuário empreendedor
        * O usuário realizou cadastro da empresa
        * O usuário escolheu uma tese de investimento
        * O sistema chamou algoritmo de análise da empresa
        * O sistema retorna para o usuário o resultado da análise e, se houver, sugestões de mudança que possam aumentar as chances de receber investimento

- Fluxo de exceção:
    - Usuário não preenche todas as informações necessárias: Sistema indica que faltam informações para enviar o cadastro, e obter análise

4.6 Estrutura e arquitetura do projeto

A arquitetura do projeto é apresentada na figura abaixo (figura 6). A definição do escopo da arquitetura e da maneira de montar os componentes se baseou nos requisitos funcionais e não funcionais mencionados previamente. O projeto pode ser dividido em quatro grandes componentes: *Front End*, Serviço API, *Database* e o Algoritmo ML. Estes componentes serão discutidos com mais detalhes na seção Desenvolvimento. O banco de dados escolhido foi a plataforma MongoDB. Os outros componentes foram desenvolvidos em *Python*, cada componente tem sua lista de bibliotecas essenciais para o seu funcionamento correto, e no diagrama foi destacada a biblioteca principal em cada ambiente.

Para o projeto proposto, o usuário interage com o sistema apenas através do *Front End*, componente responsável por fornecer as telas da página web na qual é possível inserir de dados no sistema, e tambémm analisar os resultados obtidos do mesmo.

O serviço API relaciona todos os componentes do sistema, sendo responsável por realizar os pedidos de requisição dos dados e fornecê-los ao componente solicitante. O banco de dados pode ser manipulado somente através dos endpoints fornecidos pelo serviço

API, e o algoritmo ML também se conecta com o serviço API para receber os dados que serão tratados, e armazenar na base de dados os resultados gerados.

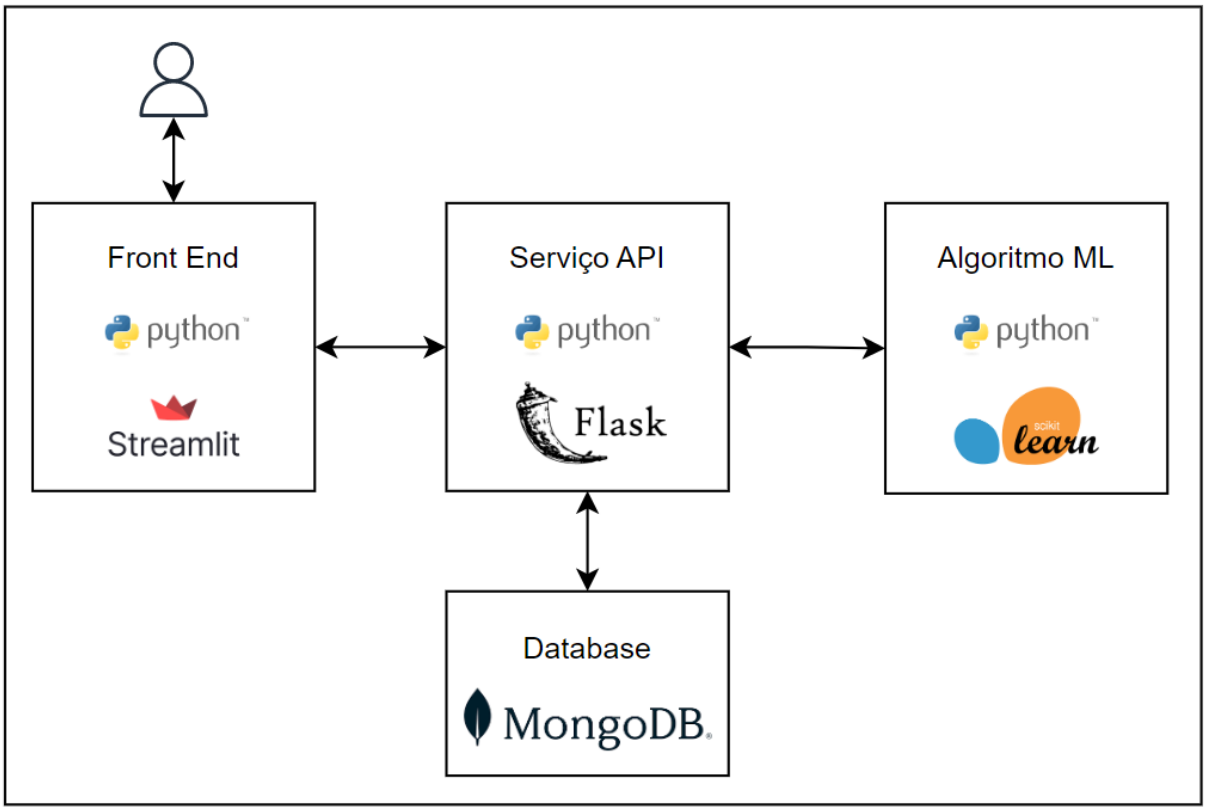


Figura 6: Arquitetura proposta para o projeto. Fonte: produzida pelos autores.

4.7 Funcionalidades de cada componente

Cada componente do sistema desenvolvido será brevemente apresentado nesta seção. Os detalhes do desenvolvimento e validação serão comentados em outras seções do trabalho. Para melhor entendimento, alguns conceitos também são explicados nessa seção,

4.7.1 Front End

Constitui a interação gráfica entre o usuário e o sistema através das telas na *web page* desenvolvida, ela disponibiliza campos para entrada de dados sobre o usuário e a empresa, e apresenta o resultado final e comentários. O Front End do sistema foi desenvolvido em *Python*, através do uso da biblioteca *Streamlit*, que através de APIs em *Python* gera arquivos em HTML para serem renderizados como páginas *Web*.

4.7.2 Algoritmo ML

Este componente realiza a modelagem do histórico de investimento de um investidor, e com base nesses modelos gerados, produz a análise de empresas. Antes da modelagem, esse componente também é responsável pela validação da base histórica, realizando a síntese e tratamento dos dados quando necessário. A síntese de dados acontece quando a base histórica fornecida ainda é muito pequena, e o tratamento de dados certifica que os dados estão em formatos compatíveis, como tipos datas e boleanos.

O algoritmo foi desenvolvido em *Python*, fortemente auxiliado pelas bibliotecas Scikit Learn e Pandas. A biblioteca Pandas é muito utilizado na área de dados, ela fornece ferramentas para trabalhar com *Dataframes*, ou seja, com tabelas que acomodam os dados analisados. A biblioteca Scikit Learn suporta técnicas de aprendizado de máquina. Através de suas funcionalidades e da base histórica é gerado um modelo de investimento, com base no qual novas empresas serão analisadas. As bases históricas são modelas por dois métodos de classificação, K-NN e Random Forest. A decisão do modelo a ser utilizado é baseada na métrica de acurácia de cada modelo.

4.7.2.1 Tese do investidor

A tese da investidor foi definida como a metodologia que um investidor usa para a tomada de decisão de investimento em empresas. Nem sempre o investidor tem conhecimento cravado sobre seus critérios, por exemplo, simplificando a tomada de decisão do investidor, ao receber propostas de empresas que estão no mercado há 1, 3, 5 e 6 anos, o investidor aprova investir nas empresas de 5 e 6 anos, mas não necessariamente sua decisão é investir em uma empresa com mais de 3 anos de mercado, pode ser que seu filtro real seja tempo de mercado mínimo de 5 anos, dessa maneira, empresas que estão há 4 anos no mercado não devem ser investidas ainda.

Para solucionar isso, o algoritmo realiza a modelagem da tese do investidor a partir da base de investimento histórica do investidor. Com base nas decisões passadas do investidor, o algoritmo é capaz de definir os critérios internamente, e os usa para identificar se uma nova empresa seria aprovada para receber investimentos. Esta tese é a modelagem K-NN ou Random Forest da base histórica, por isso, armazenar a tese é puramente salvar os dados históricos.

4.7.2.2 Dados externos

Para validação do algoritmo desenvolvido é necessário o uso de uma base de dados que simule uma base de investimentos histórica real. Essa base de dados poderia ser sintetizada com aleatoriedade, ou obtida de bases externas como o Crunchbase. Para a finalidade do nosso projeto, foi decidido simular dados de empresas e os próprios integrantes do grupo tomaram a decisão de investir ou não nas empresas criadas. Essa pequena base de dados das empresas e seus respectivos resultados foi submetida ao algoritmo desenvolvido para auxiliar na validação.

4.7.3 Back End

O Back End contempla tanto o banco de dados construído no MongoDB, como o serviço API que permite a interação com o banco de dados. Nessa seção também se discute sobre ontologia.

4.7.3.1 Serviço API e MongoDB

O banco de dados escolhido foi o MongoDB, um banco NoSQL, open source. Ele será conectado via serviço API, e será dividido em *databases* e *collections* para organizar os dados que serão armazenados.

O Back End foi desenvolvido em Python com a biblioteca Flask. Este serviço é responsável por prover os dados pedidos pelo Front End além de interagir com o algoritmo de análise, para armazenas resultados de novas análises, por exemplo. Esses endpoints se conectam com o banco de dados na plataforma MongoDB. Foram tratados alguns casos de exceção, como atualizar ou deletar um registro não encontrado no banco de dados, porém, para o desenvolvimento desse trabalho de conclusão de curso, no qual o foco é a análise de propostas de investimento, o sistema não cobre todas as situações extremas.

4.7.3.2 Ontologia

A ontologia foi a forma usada para definir o conceito *Empresa*. Na ontologia, a partir do conceito a ser definido (localizado no centro do diagrama), são relacionados aspectos ligados diretamente a ele, como Produto se relaciona com Empresa. A partir dessas "ramificações", outros conceitos vão se relacionando, como Empresa-Time-Tamanho-Quantidade de funcionários no modelo Pessoa Jurídica (PJ).

A ontologia desenvolvida nesse projeto para definir o conceito de Empresa é apresentado na figura a seguir.

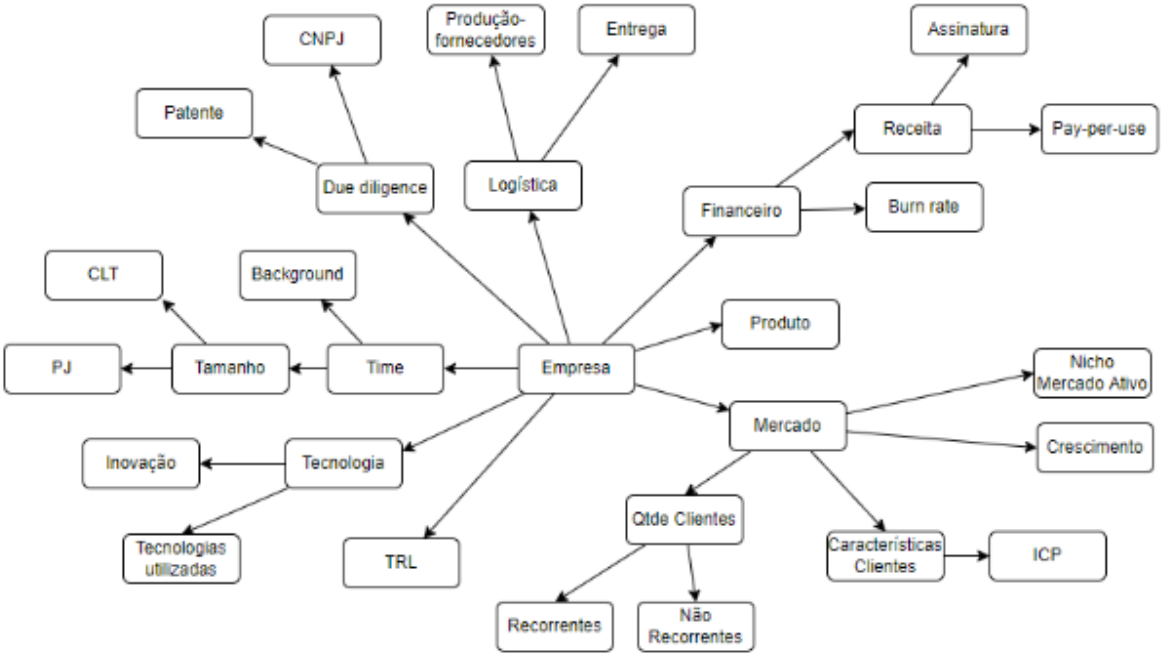


Figura 7: Ontologia do conceito *empresa*. Fonte: produzida pelos autores.

Este conceito de ontologia é relevante para adequar qualquer base histórica ou dados de uma nova empresa, ou seja, com quaisquer colunas, é possível aplicar a análise pela inteligência desenvolvida. Por exemplo, à tese do investidor pode ser mais relevante a quantidade total de funcionários, mas se a empresa fornece a quantidade de funcionários PJ, CLT e sócios, podemos automaticamente inferir o tamanho total do time, com base nas relações fornecidas pela ontologia.

5 DESENVOLVIMENTO

O grupo foi orientado a dividir as tarefas em quatro áreas: desenvolvimento do *Front End*, do *Back End*, dos algoritmos de aprendizado de máquina e do formato do relatório final.

O desenvolvimento do *Front End* envolve o planejamento das telas por onde o usuário irá interagir com o sistema. O *Back End* abrange a organização e manutenção do banco de dados, tanto das informações de empresas como os parâmetros e pesos definidos por cada investidor. Aa informações sobre as empresas serão utilizadas como base de aprendizado para as novas análises. E por fim, o desenvolvimento do algoritmo para aprendizado de máquina, sendo que a primeira versão será desenvolvida em Excel, para servir como validação da proposta. O desenvolvimento do algoritmo de classificação K-NN em Excel é abordado em uma seção mais adiante. Cada área de desenvolvimento abrange seus respectivos testes para validação. Por fim, o relatório final será pensado com base na evolução de cada parte do projeto.

Apesar da divisão de atividades em grandes áreas, o grupo todo se manteve ciente da evolução de cada área, para que as partes consigam conversar com sucesso.

5.1 Front End

Neste eixo de desenvolvimento, em conformidade com a abordagem de *mock-ups* definida pelo grupo, foi esboçado um desenho de baixa-fidelidade no papel. A partir disso, o grupo desenvolveu as telas e interação do usuário (definido como "*Front End*") utilizando a biblioteca Python Streamlit (16). O Streamlit é uma biblioteca de código aberto muito utilizada em ciência de dados, e aprendizado de máquina, para criar e compartilhar aplicativos web. Desse modo, com esta biblioteca foi bastante conveniente para integração com os demais eixos de desenvolvimento.

O formato escolhido pelo grupo para interação com o usuário foi através de formulários

(17) - um módulo nativo da biblioteca Streamlit. Entende-se que este modelo proporciona uma experiência do usuário positiva e fluida, tornando claro para os usuários o que deve ser respondido (e formatos para resposta). Outro fator positivo para a escolha dos formulários, é a conveniente formatação e tratamento dos inputs, para o armazenamento no banco de dados e para sequência dos algoritmos (através das variáveis).

Outra atividade que está nesta frente de atuação, é o levantamento das métricas que refletem o desempenho do sistema, como por exemplo o tempo de carregamento da página.

5.1.1 Streamlit: biblioteca e implementação

O fluxo desenhado para a ferramenta contempla dois usuários, conforme a descrição no capítulo anterior: investidor e empreendedor ("*founder*"). Logo, foram implementadas 4 telas:

- **Cadastro:** tela inicial da ferramenta, na qual o usuário define que fluxo seguirá (investidor ou empreendedor) e formaliza seu cadastro na ferramenta;
- **Sou Investidor:** tela de interação dos usuários investidores, na qual os usuários já estão em seu perfil e podem testar diferentes empresas na sua própria tese;
- **Sou Founder:** tela inicial de interação dos usuários empreendedores, na qual esses usuários poderão escolher, dentre as teses cadastradas na ferramenta, a qual aplicar e colher os respectivos feedbacks;
- **About:** tela informativa sobre o projeto.

5.1.1.1 Página Cadastro

A página de **Cadastro**, permite que o usuário defina qual o caso de uso desejado, como na figura 8. A partir dessa escolha, na mesma página, são exibidos os respectivos formulários de cadastro. Caso seja escolhido o fluxo do investidor, o usuário terá a opção de baixar um *template* ou um modelo de dados sintético, feito pelo grupo, conforme figura 9. Ambos os arquivos foram idealizados para facilitar a utilização da plataforma, pois ao longo do cadastro é exigido do usuário-investidor que ele forneça um histórico estruturado das suas últimas análises e decisões (o qual, servirá de base de treino dos modelos). Desse modo, o usuário poderá baixar um modelo já estruturado e preenchê-lo, ou caso deseje apenas testar a ferramenta, baixar a base sintética montada pelo grupo e seguir com seus testes. Na mesma página, o usuário-investidor poderá preencher o formulário de cadastro

com suas informações pessoais e o arquivo de seu histórico, conforme figura 10, e iniciar a análise.

Figura 8: Cabeçalho da ferramenta, onde o usuário define se é investidor ou empreendedor. Fonte: sistema desenvolvido pelos autores.

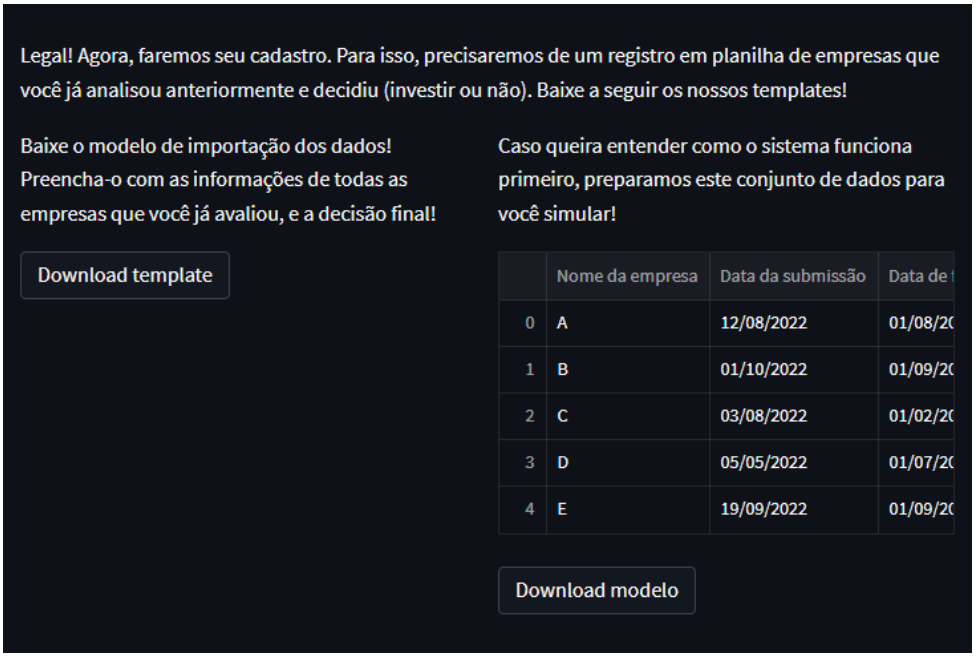


Figura 9: Template e base sintética disponíveis para download do usuário investidor. Fonte: sistema desenvolvido pelos autores.

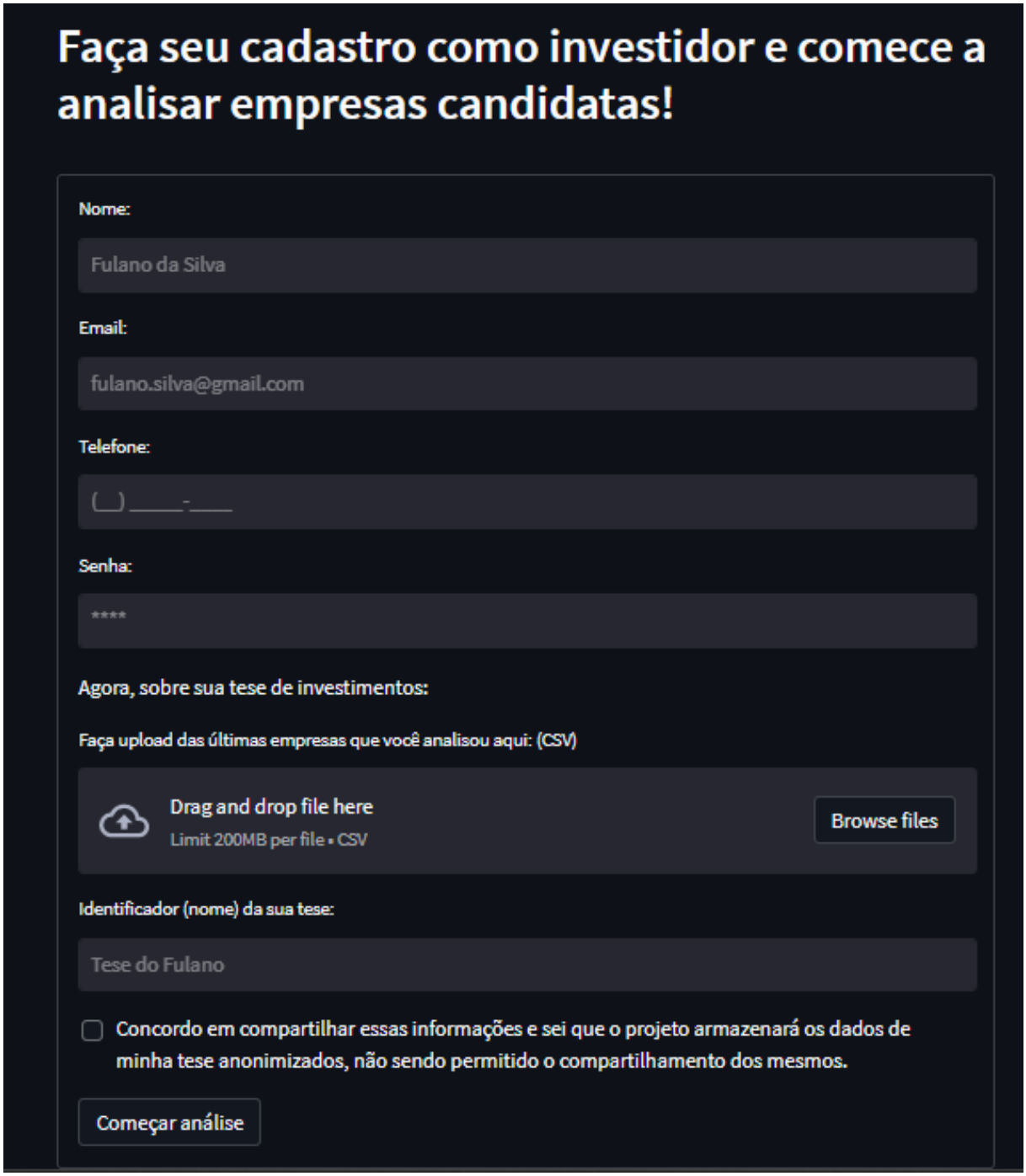


Figura 10: Formulário de cadastro do usuário investidor e sua tese. Fonte: sistema desenvolvido pelos autores.

Ao clicar no botão “Começar análise”, ainda na primeira página, a tese será submetida e se iniciará a análise da mesma. Primeiramente, é exibido o arquivo que fora enviado, figura 11, e se iniciam os processos de *Data-Augmentation*. Em seguida, se iniciam os processos de análise das variáveis e são expostas as variáveis com maior correlação com a variável-alvo (decisão “Investiu”), estas serão as variáveis a servirem de base de treino para

os modelos. Por fim, a rotina de treino dos modelos *Random Forest* e *K-Nearest Neighbors* é a última a ser conduzida na página de cadastro, ao final é feita uma comparação de qual modelo produziu a melhor acurácia num mesmo conjunto de validação e é selecionado tal modelo para que um relatório de desempenho seja exposto na tela, como na figura 12. Ao final desta tela, o usuário deverá clicar no botão "Finalizar cadastro e avaliar uma empresa" e será direcionado para a página **"Sou Investidor"**.

Figura 11: Confirmação da tese submetida e cinco variáveis mais relacionadas à decisão. Fonte: sistema desenvolvido pelos autores.

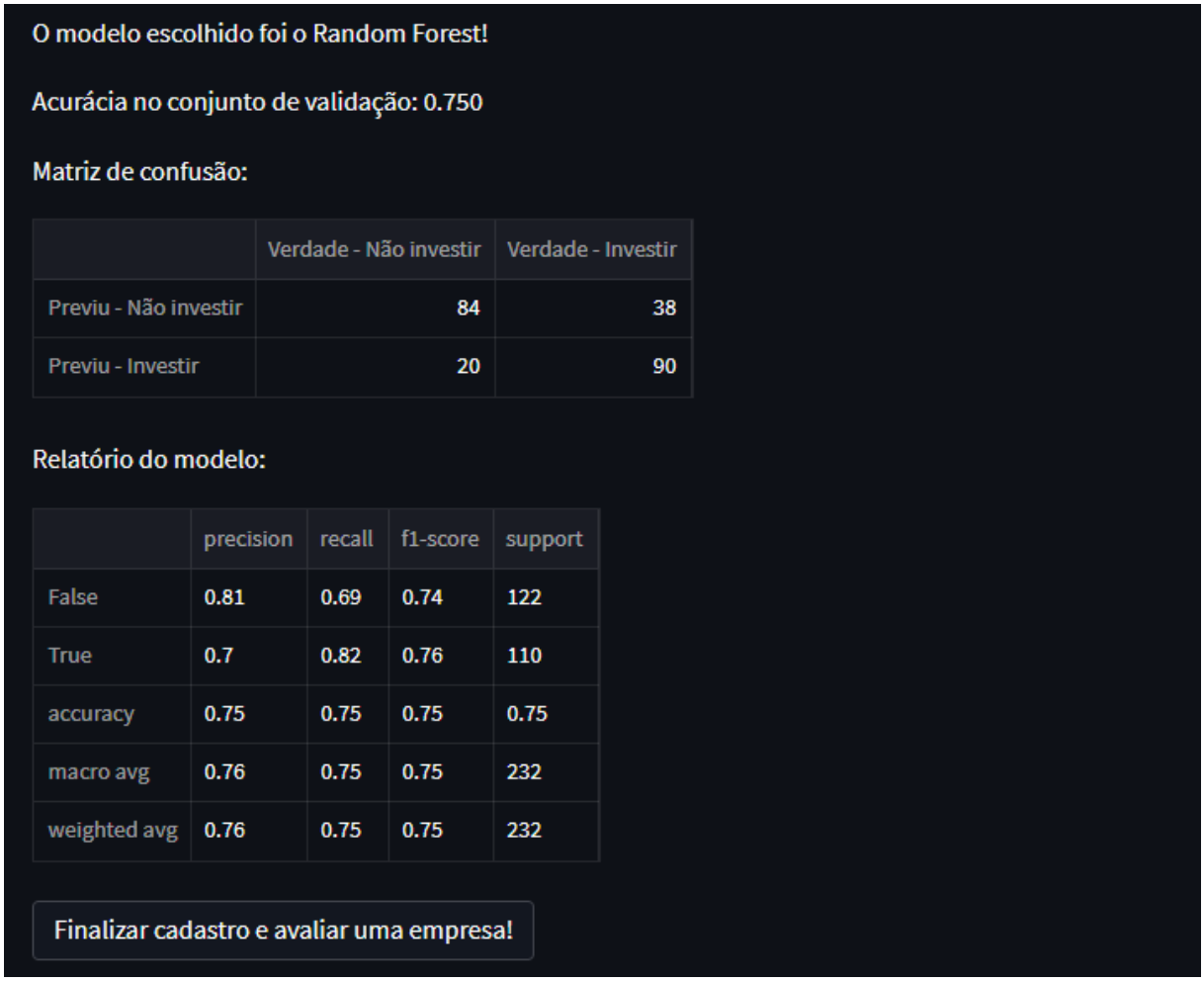
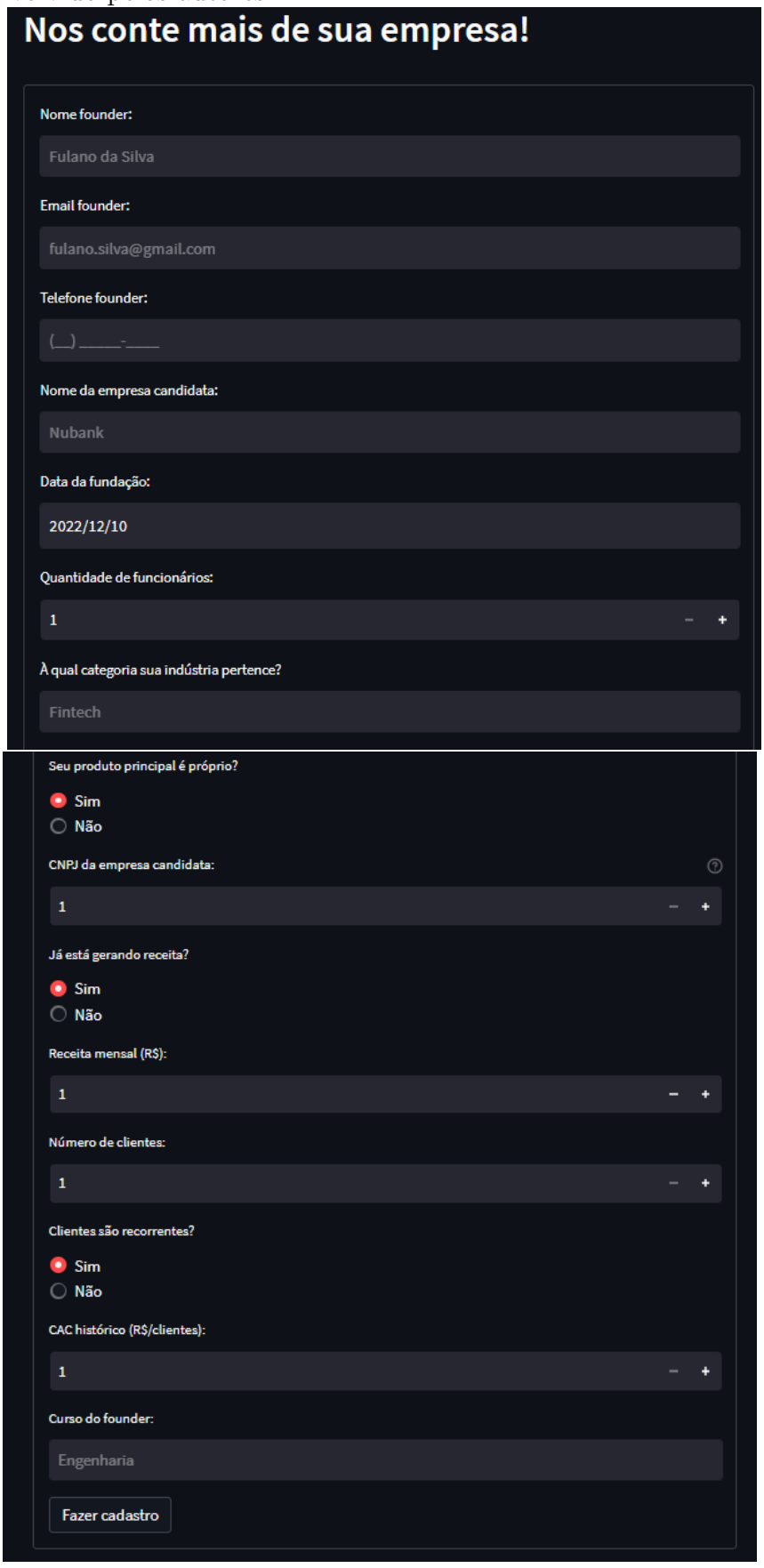


Figura 12: Seleção do modelo que performou melhor para o mesmo conjunto de dados e seu respectivo relatório de performance. Fonte: sistema desenvolvido pelos autores.

A outra opção de fluxo é caso o usuário, nesta mesma página, escolha a opção de empreendedor na figura 8. Assim como no fluxo anterior, é exigido do usuário-empreendedor o preenchimento do formulário apresentado na figura 13. Ao preencher e submeter sua empresa, clicando no botão "Fazer cadastro", o empreendedor é então direcionado para a página **"Sou Founder"**.

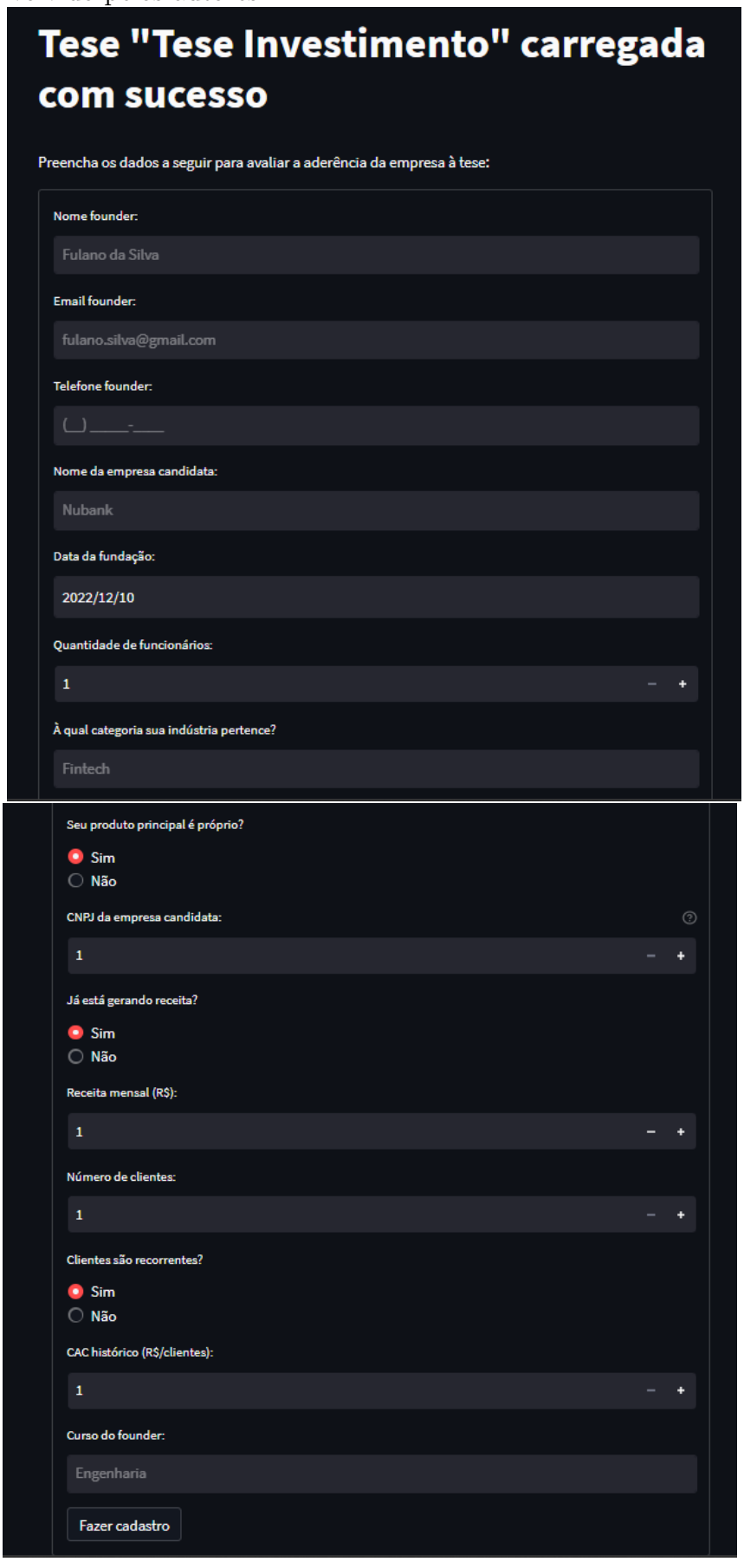
Figura 13: Formulário de cadastro do empreendedor na página de Cadastro. Fonte: sistema desenvolvido pelos autores.



5.1.1.2 Página Sou Investidor

A página **Sou Investidor** fora idealizada para ser acessada apenas pelo usuário-investidor. Desse modo o investidor é recebido por uma mensagem personalizada e poderá preencher um formulário de empresa candidata e avaliar a aderência dessa empresa a sua tese, previamente cadastrada, conforme na figura 14. Então, feito o cadastro da empresa candidata o sistema recupera a tese do usuário-investidor em questão e treina novamente os modelos de aprendizado de máquina para avaliar a empresa candidata.

Figura 14: Formulário de cadastro de empresa candidata na página do investidor. Fonte: sistema desenvolvido pelos autores.



Primeiramente, é feita a separação das variáveis mais correlacionadas da empresa candidata (de acordo com o que fora apresentado na página anterior), conforme figura 15, e o treino dos modelos.

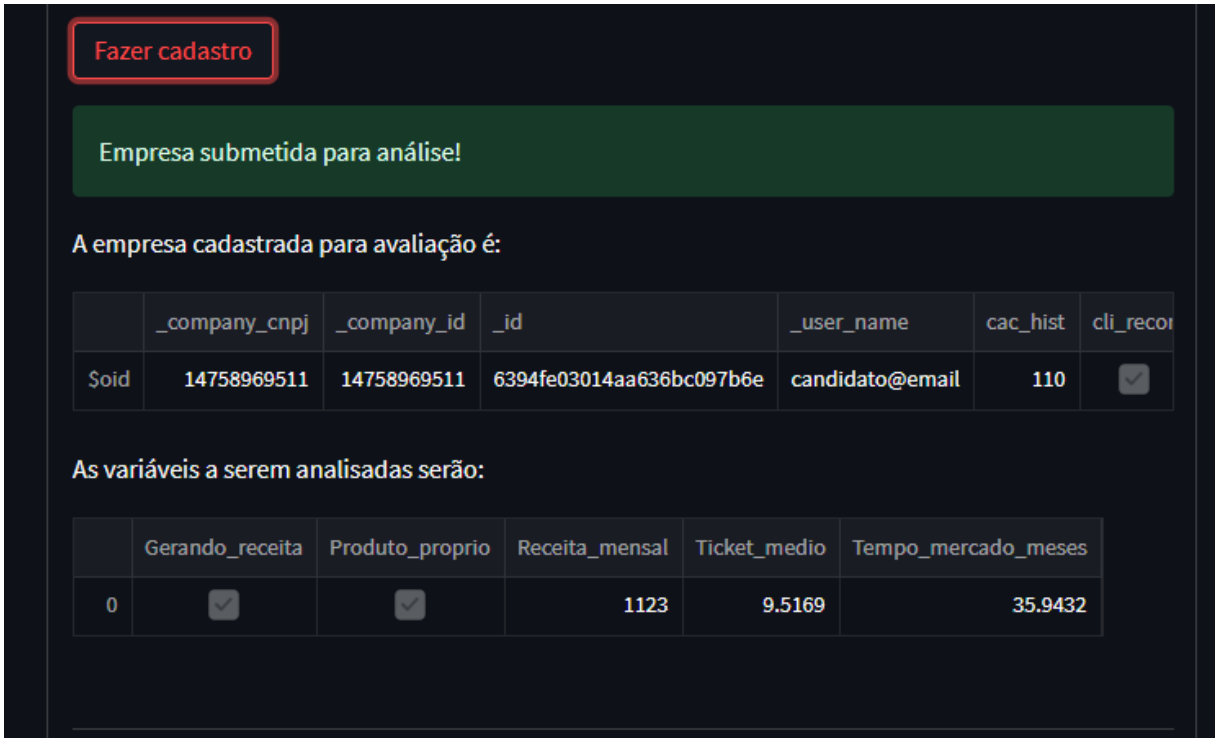


Figura 15: Confirmação de cadastro de uma empresa candidata para uma tese de investimento e as variáveis que serão analisadas. Fonte: sistema desenvolvido pelos autores.

Tendo em vista o exercício de demonstração, o grupo optou por manter nesta versão da ferramenta ambos os modelos e seus relatórios na página do investidor, de modo que tanto o *Random Forest* quanto o *K-Nearest Neighbors* são utilizados para fazer a classificação. A análise então é exposta de acordo com uma sequência lógica:

- **Modelo:** qual modelo está devolvendo a análise;
- **Decisão:** a classificação resultante do método de predição do modelo, junto a uma mensagem padrão de acordo com essa classificação;
  - **Análise:** caso a classificação seja **não investir**, inicia uma rotina de recuperação do exemplo mais próximo da empresa candidata, e informa isso ao usuário;
  - **Info:** caso a classificação seja **não investir**, retorna propostas de alterações nos parâmetros, dos quais é possível inferir o que fora determinante na eliminação da empresa candidata no processo.

Dessa forma, nesta página o usuário investidor terá acesso às classificações de cada modelo, sejam elas decisões positivas ou negativas, respectivamente representadas nas figuras 16 e 17.

Figura 16: Exemplo de empresa candidata aprovada na análise de ambos os modelos.Fonte: sistema desenvolvido pelos autores.

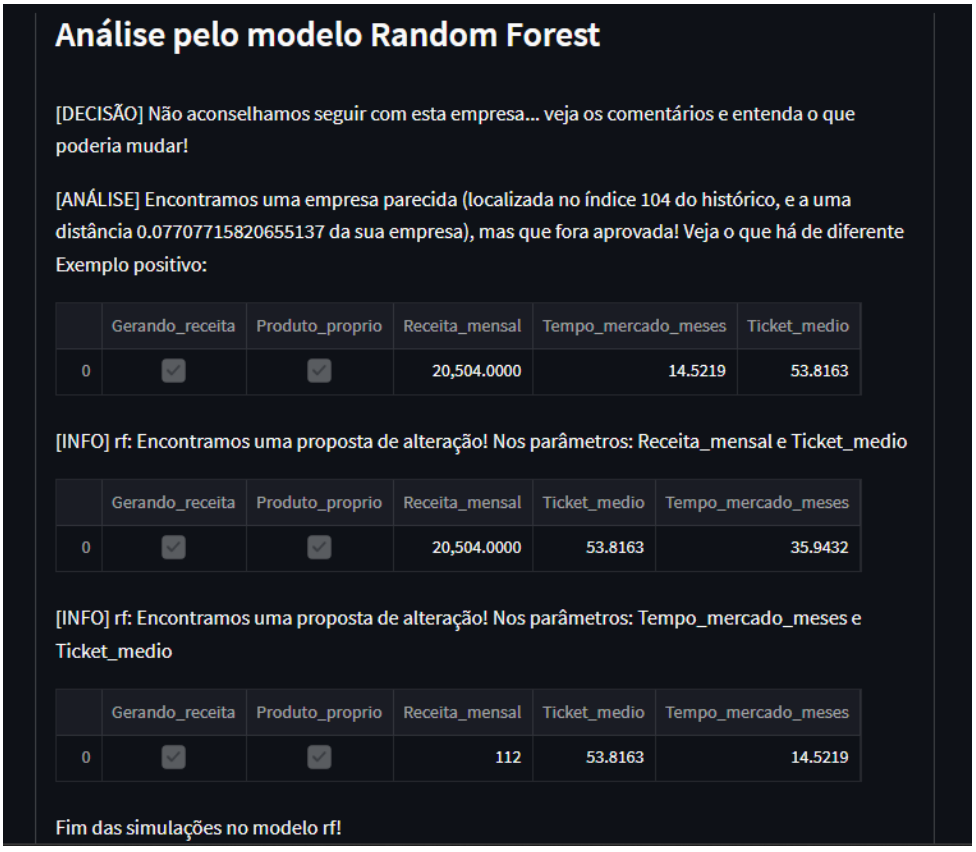


Figura 17: Exemplo de empresa candidata reprovada pelo modelo *Random Forest*, e as sugestões de alteração nos parâmetros. Fonte: sistema desenvolvido pelos autores.

5.1.1.3 Página Sou Founder

Feito o cadastro do empreendedor na página inicial, o usuário-empreendedor é direcionado para esta página e terá a oportunidade de escolher qual investidor, e sua respectiva tese, deseja aplicar sua empresa. No primeiro momento, é possível conferir breves informações da tese de investimento, relacionadas à identificação do investidor e apenas ao clicar no botão "Começar a análise da empresa de acordo com a tese ¡*Nome da Tese*" é iniciada a análise, conforme se verifica na figura 18. Ao clicar, se inicia um processo de análise análogo ao que fora desenvolvido e explicado na seção da página "Sou Investidor".

Figura 18: Página do empreendedor para que escolha qual investidor e respectiva tese irá se candidatar. Fonte: sistema desenvolvido pelos autores.

5.1.1.4 Página About

A página ***About*** se refere a uma breve descrição do projeto, seus objetivos e referência à página do projeto, conforme figura 19.

Figura 19: Página "About" da ferramenta. Fonte: sistema desenvolvido pelos autores.

5.2 Serviço API e Banco de Dados

O banco de dados usado nesse projeto foi construído na plataforma MongoDB, uma plataforma NoSQL. E a sua alteração é feita exclusivamente pelo serviço API desenvolvido, favorecendo positivamente na questão de segurança dos dados. Nas subseções a seguir, cada componente será discutido com mais detalhes.

5.2.0.1 Serviço API

O serviço API foi desenvolvido em Python usando a biblioteca Flask, ele fornece *endpoints* para interação com o banco de dados. Os endpoints suportam as operações básicas CRUD (*Create*, *Read*, *Update* e *Delete*, ou seja, criação, leitura, atualização e remoção dos dados). Os *endpoints* desenvolvidos para esse projeto fazem algumas validações básicas como verificação da existência de registro antes de tentar alterá-lo, para levantar alertas ao usuário e evitar erros que possam quebrar a aplicação, mas na maioria dos casos foi considerado que o usuário só realiza operações válidas e corretas, sendo desnecessários o tratamento completo de todas as possíveis exceções. Para aplicar no mercado é de extrema relevância a administração desses casos que possam causar erros.

O código do serviço em Python está disponível no repositório do GitHub, e seu arquivo README.md possui algumas instruções para rodá-lo localmente. São exibidos também alguns comandos como sugestão para testar os endpoints com o serviço rodando.

O serviço API em Python e o banco MongoDB poderiam ser deployados em algum serviço como Amazon AWS ou Google Cloud Platform, mas para os objetivos desse trabalho, os serviços são rodados localmente e possibilitam realizar nossas análises de maneira satisfatória.

Para testar os endpoints, o serviço foi rodado localmente, figura 20, e as ordens foram dadas por linha de comando para cada operação, como exemplificado nas figuras 22, 24, 26 e 29. Os arquivos do tipo JSON também estão no repositório, contendo alguns exemplos de conteúdo que podem ser usados nos testes. Vale lembrar que nem todos os endpoints tem o tratamento de erros completo, por isso vale conferir se o conteúdo do arquivo JSON está coerente com o endpoint que deseja testar, por exemplo, certifique que o registro em arquivos modelos para operações de inserção devem ser registros com o *userid* (identificador do usuário) único, ainda não existente na base.

```
(venv_tcc) C:\Users\User207\Documents\TCC\service>python pymongo_test.py
 * Serving Flask app 'pymongo_test' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead
.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 377-574-867
```

Figura 20: Serviço API rodando localmente. Fonte: sistema desenvolvido pelos autores.

Para efeitos de comprovação de funcionamento, alguns testes foram realizados e são indicados a seguir. Os testes foram feitos com a base de usuários, testando todas as operações disponibilizadas.

```
(venv_tcc) C:\Users\User207\Documents\TCC\service>python pymongo_test.py
 * Serving Flask app 'pymongo_test' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead
.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 377-574-867
127.0.0.1 - - [11/Dec/2022 17:05:03] "POST /profile HTTP/1.1" 200 -
127.0.0.1 - - [11/Dec/2022 17:05:11] "GET /profile/cmi_user HTTP/1.1" 200 -
127.0.0.1 - - [11/Dec/2022 17:05:59] "PUT /profile/cmi_user HTTP/1.1" 200 -
127.0.0.1 - - [11/Dec/2022 17:06:06] "DELETE /profile/cmi_user HTTP/1.1" 200 -
```

Figura 21: Serviço API rodando localmente para realização de testes locais e apresentando a resposta obtida em cada solicitação recebida. Fonte: sistema desenvolvido pelos autores.

O primeiro teste envia uma solicitação de inserção de registro no banco de dados do usuário.

```
(venv_tcc) C:\Users\User207\Documents\TCC\service\json_files>curl -i -H "Content-Type: application/json" -X POST -d @
post_profile.json http://localhost:5000/profile
HTTP/1.1 200 OK
Server: Werkzeug/2.2.2 Python/3.10.2
Date: Sun, 11 Dec 2022 20:05:03 GMT
Content-Type: application/json
Content-Length: 187
Connection: close

{
  "_id": {
    "$oid": "639637efa45b7fb44b14833e"
  },
  "_user_name": "cmi_user",
  "email": "cmi@email.com",
  "investor": true,
  "name": "Camila M I",
  "password": "cmi123"
}
```

Figura 22: Teste com endpoint para inserir (POST) um elemento usuário do sistema. Fonte: sistema desenvolvido pelos autores.

O conteúdo do arquivo JSON enviado é mostrado a seguir:

```
{
    "_user_name": "cmi_user",
    "name" : "Camila M I",
    "email" : "cmi@email.com",
    "investor": true,
    "password": "cmi123"
}
```

Figura 23: Conteúdo do arquivo enviado para a operação de inserção de registro de usuário. Fonte: sistema desenvolvido pelos autores.

Para obter um determinado registro, é necessário identificar o registro procurado, para o endpoint GET do usuário, é necessário definir o *_user_id*. O teste abaixo procura pelo registro adicionado na operação mencionada acima, portanto procura pelo valor "cmi_user" como *_user_id*:

```
(venv_tcc) C:\Users\User207\Documents\TCC\service\json_files>curl -i -X GET http://localhost:5000/profile/cmi_user
HTTP/1.1 200 OK
Server: Werkzeug/2.2.2 Python/3.10.2
Date: Sun, 11 Dec 2022 20:05:11 GMT
Content-Type: application/json
Content-Length: 187
Connection: close

{
  "_id": {
    "$oid": "639637efa45b7fb44b14833e"
  },
  "_user_name": "cmi_user",
  "email": "cmi@email.com",
  "investor": true,
  "name": "Camila M I",
  "password": "cmi123"
}
```

Figura 24: Teste com endpoint para leitura (GET) um elemento usuário do sistema. Fonte: sistema desenvolvido pelos autores.

O mesmo registro também pode ser verificado na plataforma web do MongoDB, conforme figura 25:

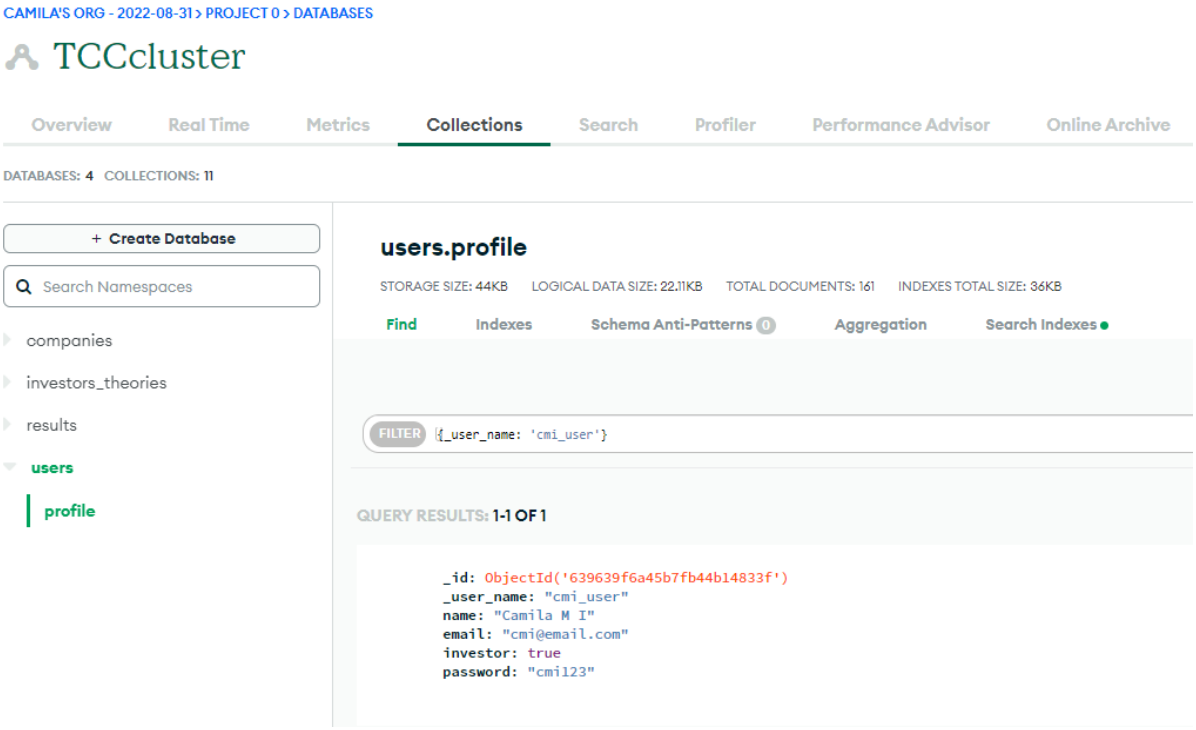


Figura 25: Registro inserido e verificado na plataforma web MongoDB. Fonte: sistema desenvolvido pelos autores.

Continuando com os testes locais com o serviço API. Para atualizar um registro, é necessário informar o identificador no endpoint e enviar um objeto JSON contendo as informações a serem atualizadas. A lógica no endpoint contém a validação de que o registro a ser removido existe no banco de dados.

```
(venv_tcc) C:\Users\User207\Documents\TCC\service\json_files>curl -i -H "Content-Type: application/json" -X PUT -d @p
ost_profile.json http://localhost:5000/profile/cmi_user
HTTP/1.1 200 OK
Server: Werkzeug/2.2.2 Python/3.10.2
Date: Sun, 11 Dec 2022 20:05:59 GMT
Content-Type: application/json
Content-Length: 187
Connection: close

{
  "_id": {
    "$oid": "639637efa45b7fb44b14833e"
  },
  "_user_name": "cmi_user",
  "email": "cmi@email.com",
  "investor": true,
  "name": "Camila M I",
  "password": "cmi123"
}
```

Figura 26: Teste com endpoint para atualizar (PUT) um elemento usuário do sistema. Fonte: sistema desenvolvido pelos autores.

O arquivo JSON enviado precisa conter alguma chave de atualização (Field Update Operators (18)), no caso apresentado a intenção é definir valores novos para campos já

existentes, por isso o uso da chave *$set*.

```
{
    "$set":
    {
        "name" : "Camila Miwa Ivano",
        "password": "cmi456"
    }
}
```

Figura 27: Conteúdo do arquivo enviado para a operação de atualização de registro de usuário. Fonte: sistema desenvolvido pelos autores.

O objeto atualizado já pode ser validado na plataforma web MongoDB:

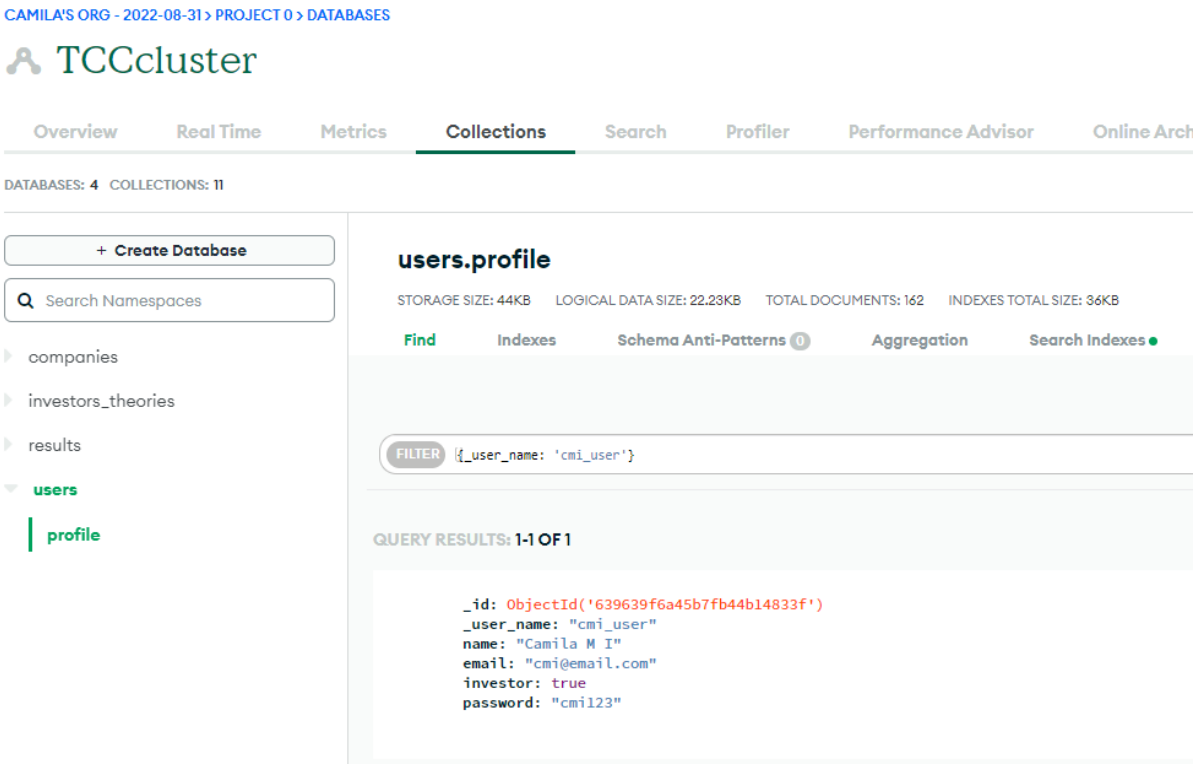


Figura 28: Registro do usuário atualizado no banco de dados. Fonte: sistema desenvolvido pelos autores.

Para deletar um registro, é necessário informar o identificador no endpoint. A lógica no endpoint contém a validação de que o registro a ser removido existe no banco de dados.

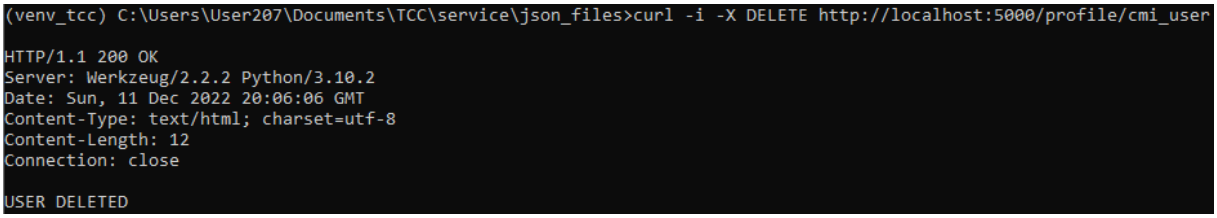


Figura 29: Teste com endpoint para deletar (DELETE) um elemento usuário do sistema. Fonte: sistema desenvolvido pelos autores.

A estrutura dos demais endpoints seguem a mesma padronização. Portanto os testes seriam análogos,as variações estão relacionadas com os identificadores exigidos por cada endpoint.

5.2.1 MongoDB

O MongoDB foi escolhido para armazenar os dados coletados e gerados pelo sistema, são eles od dados dos usuários empreendedores e investidores, das empresas a serem analisadas, das bases históricas de investimento (que fornecem os insumos para a modelagem das teses), e resultados das análises com sugestões de modificação. O MongoDB é um banco de dados NoSQL e essa característica se mostrou bastante vantajosa para o projeto discutido.

A vantagem de um banco de dados NoSQL em relação à tabelas SQL, é que o modelo NoSQL permite o armazenamento de dados como objetos JSON, sem muitas restrições, por exemplo, em tabelas SQL são definidas as colunas de cada tabela, e para a inserção de novos registros é necessário a definição de todas as colunas nesse novo objeto, mesmo que seja uma célula vazia, ou seja, cada inserção nova no banco exige o conhecimento prévio das colunas configuradas. Em bancos NoSQL, como o MongoDB, cada objeto pode conter colunas diferentes, se tornando mais flexível. Para exemplificar essa vantagem, considere que duas empresas realizam seu cadastro informando o número total de funcionários, uma delas pode identificar esse parâmetro como “Quantidade de Funcionários total” e outra pode nomear como “Total de funcionários”. Num banco SQL, onde as colunas tem seus nomes já definidos, se as inserções não respeitarem o nome pré determinado, o cadastro gera erro, mas no MongoDB não. O tratamento das colunas para identificar que ambas as colunas expressam a mesma característica é de responsabilidade do algoritmo.

Como comentado, o MongoDB aceita maior flexibilidade dos dados armazenados, por ser do tipo NoSQL, assim bases de dados diversas podem ser armazenadas sem problemas,

passando para o algoritmo a responsabiliade de tratamento desses diversos dados.

A estrtura do MongoDB possibilita a criação de um *Cluster* (veja detalhes da implementação na figura 30) que agrega todos os *Databases*, que por sua vez reúne todas as *Collections*. Para a organização, foi criado um único *Cluster*, no qual foram gerados *Databases*, um para cada finalidades, por exemplo, um para dados relativos aos usuários, um segundo para dados sobre as empresas a serem avaliadas, outro para dados das bases históricas, e por fim outro para os resultados obtidos de cada análise. E dentro de cada *Database*, organizamos em *Collections* cada conjunto de dados. Essa segregação em Databases Collections é melhor para a manutenção e disponibilidade do sistema, pois evita que falhas em um conjunto específico de dados acabe prejudicando os outros conjuntos.

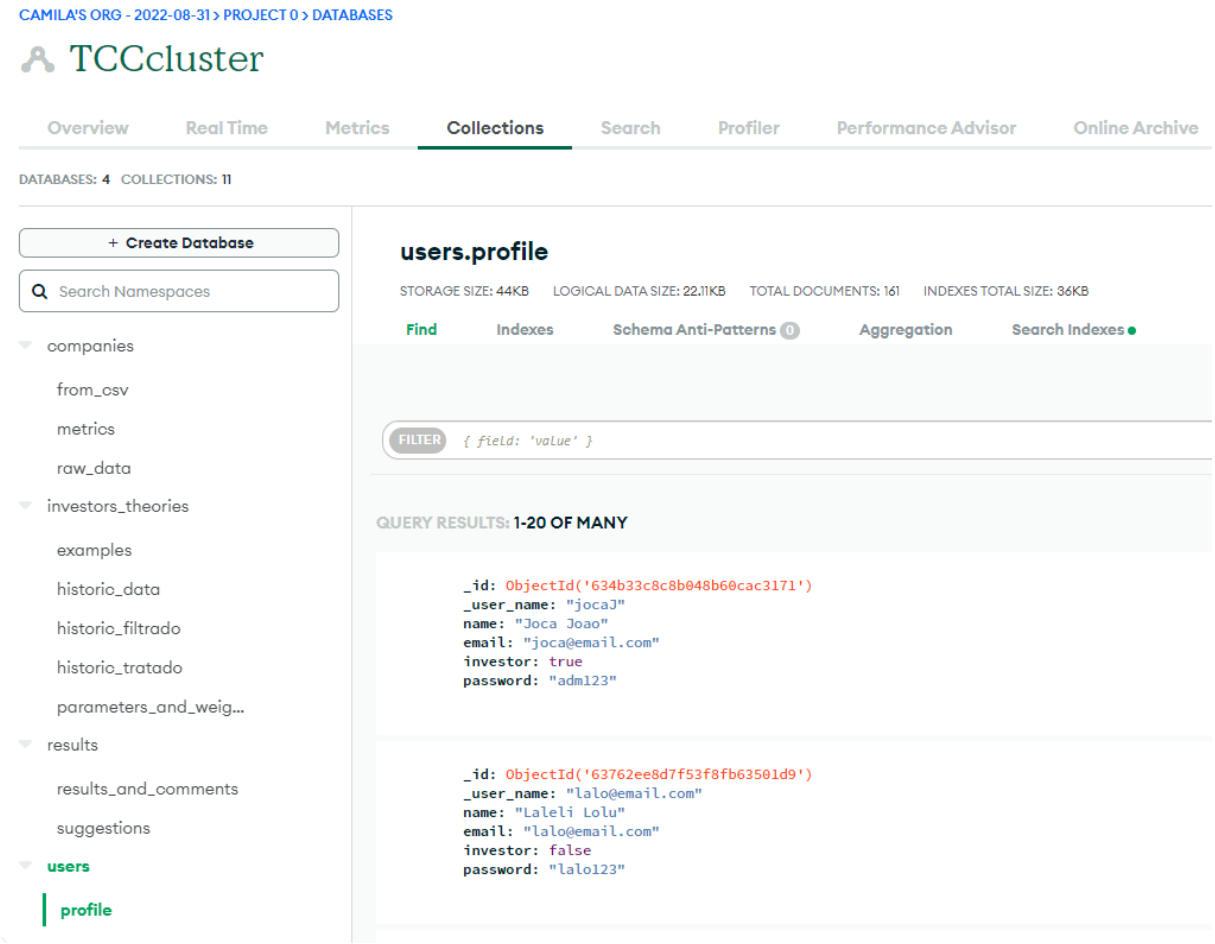


Figura 30: Estrutura desenvolvida no MongoDB. Fonte: sistema desenvolvido pelos autores.

Como já mencionado, o MongoDB aceita registros em formato JSON, na figura 31 é possível analisar um exemplo de registro no banco de dados dedicado ao armazenamento de informações relacionadas a resultados de análises. Este registro indica o resultado

da análise da empresa cujo cnpj vale “jocaJ” para a tese cujo nome é “jocaJ” (estes registros são exemplos, e por isso aparecem com nomes fictícios). O campo “approved” boleano insdica se o algoritmo foi aprovado pelo algoritmo para receber investimentos. E as sugestões obtidas são detalhadas no dicionários contidos nos campos com o nome da alteração dentro dos campos de sugestão.

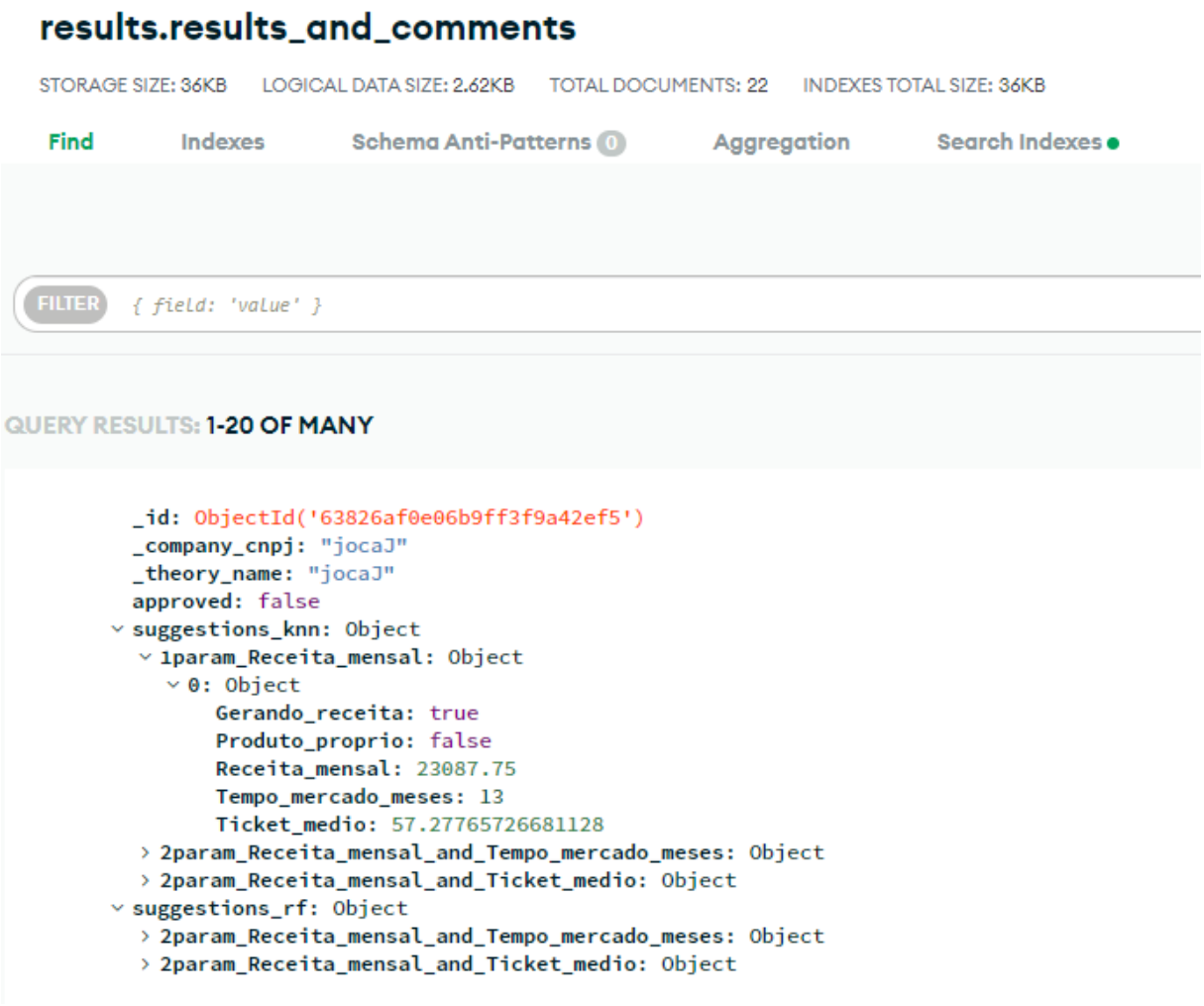


Figura 31: Exemplo de resultado armazenado no banco de dados. Fonte: sistema desenvolvido pelos autores.

5.3 Algoritmo de análise da empresa candidata

Para este eixo de desenvolvimento, alinhada à estratégia do grupo de progresso iterativo a partir de rascunhos de baixa fidelidade, foi implementado o algoritmo de classificação K-Nearest Neighbors numa planilha *Excel* com as etapas e cálculos utilizados didaticamente expostos, conforme a figura 32.

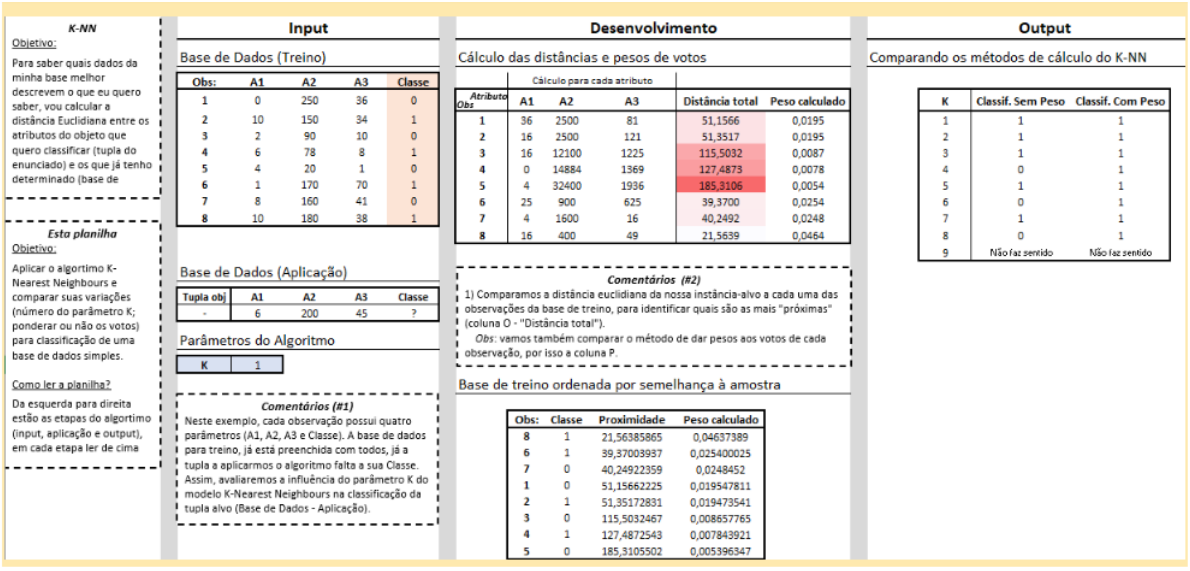


Figura 32: Print do modelo em *Excel* para a aplicação do algoritmo de classificação K-NN. Fonte: produzida pelos autores.

A partir da figura 32, destaca-se os dados de entrada necessários para o treinamento do algoritmo, isto é, uma base de treino estruturada e uma amostra candidata, de acordo com essa estrutura prévia. Entende-se ainda que os atributos A1; A2; A3 representem atributos alinhados com a organização da ontologia supracitada. O modelo em planilha serviu de insumo para o progresso da frente de algoritmos, conduzida de acordo com o *Data Science Pipeline* (11), figura 1.

Adiante, define-se o seguinte percurso para a condução de projetos de ciência de dados(11):

1. **Entendimento do problema e definição do escopo**: Caracterização do limite do problema a ser resolvido pela solução, muito importante para seleção das técnicas e algoritmos corretos de acordo com a necessidade. Neste trabalho, era um problema de classificação através de algoritmos supervisionados;

2. **Definição das métricas de sucesso:** Declaração dos indicadores que serão avaliados e do que é bom ou ruim para o resultado do modelo a ser construído, servindo de guia para a decisão de parada do processo de desenvolvimento. Aqui, entende-se como as métricas mais preciosas o *recall* e a acurácia dos modelos;

3. **Definição dos dados necessários:** Levantamento dos dados que podem ajudar na criação da solução, sejam obrigatórios ou opcionais, disponíveis atualmente ou não. Neste trabalho, fora definido um subconjunto da ontologia;

4. **Aquisição de dados:** Obtenção dos dados necessários, através dos *stakeholders* - usuários interessados no serviço - ou métodos próprios, como a sintetização de dados desenvolvida neste trabalho;

5. **Pré-processamento de dados:** Tratamento inicial dos dados para prepará-los para utilização. Um exemplo de tratamento é a remoção de nulos e de outliers;

6. **Análise exploratória de dados (E.D.A.):** Entendimento dos dados e análise das relações entre cada informação disponível;

7. ***Feature engineering:*** Criação de variáveis explicativas (*features*) que trazem mais informação para o modelo criado - manipulação dos dados para formatos melhor aproveitados pelos modelos;

8. **Construção e avaliação do modelo:** Criação, de fato, dos modelos e avaliação dos resultados obtidos. Esse passo inclui um ciclo próprio de otimização dos hiperparâmetros do modelo;

9. **Comunicação dos resultados:** Apresentação dos resultados obtidos de forma que todos os envolvidos tenham conhecimento da qualidade e das falhas dos modelos. Ainda mais importante a comunicação aos envolvidos de perfil não técnico, sem conhecimento de ciência de dados e intrínseco à natureza do recorte estudado neste trabalho;

10. **Implantação:** Disponibilização do modelo em produção, ou seja, incluí-lo no processo que fará uso do resultado do modelo;

11. **Monitoramento e Manutenção:** Acompanhamento dos resultados do modelo em produção, verificando se estão de acordo com o esperado ou se há desvios. Se necessário, fazer ajustes e retreinar o modelo (passar novamente por todo o ciclo).

Assim, o grupo conduziu o desenvolvimento desta frente através do ambiente *Google Colaboratory*, que permitiu trabalho conjunto e documentação apropriada em formato explicativo de *notebook*. A seguir os módulos implementados neste *notebook*:

1. Preparação das bibliotecas e importações

2. Aquisição dos dados

   (a) *Data Augmentation*

3. Preparação e limpeza dos dados
4. Análise exploratória dos dados (E.D.A.)
5. Engenharia de variáveis
6. Treino dos modelos
   (a) *K-Nearest Neighbors* (K-NN)
   (b) *Random Forest* (RF)
7. Aplicação dos modelos
8. Consolidação dos resultados

5.3.1 *Data Augmentation*

O grupo adotou a estratégia de sintetizar os dados para a demonstração da ferramenta, em virtude do caráter sigiloso das informações necessárias para um teste apropriado. Logo, estudou-se uma rotina de sintetização de dados estruturados que culminou na seguinte estrutura:

1. Ampliação da amostra
2. Filtragem dos dados sintéticos
3. Análise da variável-alvo

O primeiro passo é o fornecimento de uma base enxuta, também sintética e de autoria dos autores. Tal base inicial, como relatado na seção de resultados a seguir, se provou insuficiente para os treinos, de modo que não viabilizava as divisões dos conjuntos de maneira apropriada.

Dessa forma, fora gerada uma nova base randômica a partir da função *make regression* da biblioteca *scikit learn* (19), gerando um novo conjunto de 1000 amostras, porém pouco expressivo do problema a ser tratado, exigindo uma rotina de adaptação. Assim, o próximo objetivo consiste em atribuir as características explicativas da base enxuta à nova base, sintética. As escalas trabalhadas foram: 8 amostras na primeira base, e 1000 na segunda. A adaptação fora feita, principalmente, através do método *MinMaxScaler*(20), implementado pela biblioteca *SciKitLearn* da seguinte maneira:

$X_{std} = \frac{X - X_{min(axis=0)}}{X_max(axis=0) - X_{min(axis=0)}}$

$X_{novo} = X_{std} \times (max - min) + min$

Dessa forma, o novo conjunto da variável X, estará contido dentro do intervalo de cada variável explicativa original - isto é, da base menor.

Após essa adaptação, o grupo implementou uma rotina de limpeza da base. Como esperado, apesar de adaptado ao contexto estudado (por exemplo, as variáveis estavam de acordo com os respectivos tipos apropriados: datas no formato de *string* e formatadas coerentemente), havia algumas incoerências na base que poderiam atrapalhar a predição dos modelos. Além da filtragem dos dados, e consequentemente da redução do tamanho da base, se fez necessária uma análise mais profunda na variável-alvo, justamente para influir uma correlação a ser descoberta pelos modelos. Assim, o grupo combinou duas estratégias: tanto regras de negócio previamente estabelecidas (e detalhadas na seção de resultados) quanto uma análise manual de uma fração da base final. Assim, culminando numa base muito mais significativa e coerente com um problema de escala real.

5.3.2 *Data Science Pipeline*

A partir de uma base de dados suficientemente representativa, os próximos passos são comuns e fazem parte do processo do *Data Science Pipeline*, imagem 1. Em projetos de apresentação de um estudo de dados, é comum que todas as etapas sejam extremamente específicas ao contexto do problema e objetivos da análise. No entanto, tendo em vista que a ferramenta deste trabalho tem como objetivo atender a diferentes usuários, este ciclo fora estruturado de maneira parametrizável e fácil de agregar novas visualizações em trabalhos futuros.

A etapa de **Limpeza de dados**, se refere ao tratamento usual de desconsiderar valores nulos no modelo, avaliação do tipo das variáveis a serem trabalhadas (por exemplo, conversão das datas recebidas em *string* para o formato *datetime*) e renomear as colunas de maneira apropriada (sem caractere especial nem espaços dentro do nome). Nesta etapa de preparação, também fora realizado o método de tratamento das variáveis categóricas, o ***One-Hot Encoding***, implementado pelo método *get dummies* da biblioteca *Pandas* (21), o qual consiste em separar uma variável categórica em valores numéricos. Por exemplo, a variável "Indústria" cujos valores poderiam ser "Tech" ou "Fintech", será transformada em duas novas variáveis booleana: "Tech" e "Fintech". Essas novas variáveis são preenchidas de acordo com a variável original, "Indústria", isto é, caso "Indústria"

fosse “Tech” então a variável “Tech” terá valor “1” e a “Fintech” valor “0”. Ao final do processo, como a informação está presente no conjunto a partir dessas novas variáveis, é possível-descartar a coluna original categórica.

A etapa **Análise exploratória de dados**, é intrinsecamente relacionada à exploração do conjunto recebido. No caso deste trabalho, esse escopo fora contemplado na etapa anterior de ***Data Augmentation***. No entanto, destaca-se aqui uma rotina essencial de levantamento das variáveis mais correlacionadas à variável-alvo, pelo método *corr* da biblioteca *Pandas*, (22). O método devolve valores representativos da correlação numa escala de [-1, 1], permitindo uma análise de quais variáveis podem ser mais explicativas da decisão final, informação essencial no passo a seguir.

Já na etapa de **Engenharia de variáveis**, são feitas manipulações sobre as variáveis existentes de forma a resumir a informação presente e condensá-la em variáveis mais explicativas. Assim, podem ser criadas novas variáveis a partir das já existentes, por exemplo as variáveis “Ticket Médio” e “Tempo de mercado” sendo respectivamente: a divisão da “Receita mensal” pelo “Número de clientes”; e a quantidade de meses entre as datas “Data de submissão” e “Data de fundação”. Ao final destas criações, o modelo trabalhado possui 19 variáveis. Avalia-se, conforme discutido na seção de resultados, que é um número expressivo e se implementa uma rotina de seleção dessas variáveis: o algoritmo de eliminação recursiva de variaveis baseado em *cross-validation*, método *RFECV* (23) da biblioteca *SciKitLearn*. Este método recebe um modelo, no caso o *Random Forest*, para simulação e cria uma quantidade pré-determinada de instâncias do mesmo, de modo que a cada interação seja calculada a importância de cada variável e eliminadas as menos explicativas. Ao final do método, se avalia qual iteração obteve a melhor performance. Concluindo esta etapa, considera-se que os dados estão suficientemente estudados e manipulados para servirem de base para os modelos.

5.3.3 Treino dos Modelos

5.3.3.1 Separação dos conjuntos de dados - treino, validação e teste

Para treinar qualquer modelo de aprendizado de máquina é necessário ter definidos os conjuntos de dados a serem utilizados e seu escopo. Assim, a partir da base de dados única são divididos os três conjuntos a seguir:

1. **Conjunto de treino:** amostras usadas para realizar o *fit* dos modelos, ou seja, o modelo estudará esse conjunto de dados e procurará refletir as tendências observadas

aqui para refinar seus parâmetros;

2. **Conjunto de validação:** amostras para avaliação avaliação não enviesada dos modelos treinados a partir do conjunto de treino. Tem como objetivo a seleção do melhor algoritmo, combinação de hiperparâmetros e *features*.

3. **Conjunto de teste:** amostras usadas para providenciar uma avaliação não enviesada do modelo final treinado no dataset de treino. Este conjunto visa avaliar as definições dos hiperparâmetros do conjunto anterior.

A partir dessa divisão de funções de cada subconjunto de dados, é possível otimizar tal tarefa a partir dos métodos de validação-cruzada, cujo objetivo é mitigar os efeitos de uma eventual divisão pouco eficiente (isto é, exclusão de dados importantes para treino). Essa estratégia, fará combinações entre os conjuntos de teste e validação, de modo que cada observação seja usada tanto para treino quanto validação. Uma implementação simples deste conceito é identificada pelo método ***K-Folds***, figura 33. Este método consiste em dividir todas as obsverções em K grupos (*folds*) aleatórios, treinar o modelo em K-1 grupos e validar no K-ésimo grupo - repetindo o processo até que todos os grupos sejam utilizados como validação. A avaliação final do modelo será a média da métrica de avaliação sobre todos os k modelos treinados. Finalmente, conclui-se que a estrutura de divisão dos subconjuntos, somada às rotinas de validação cruzada, está representada na figura 34.

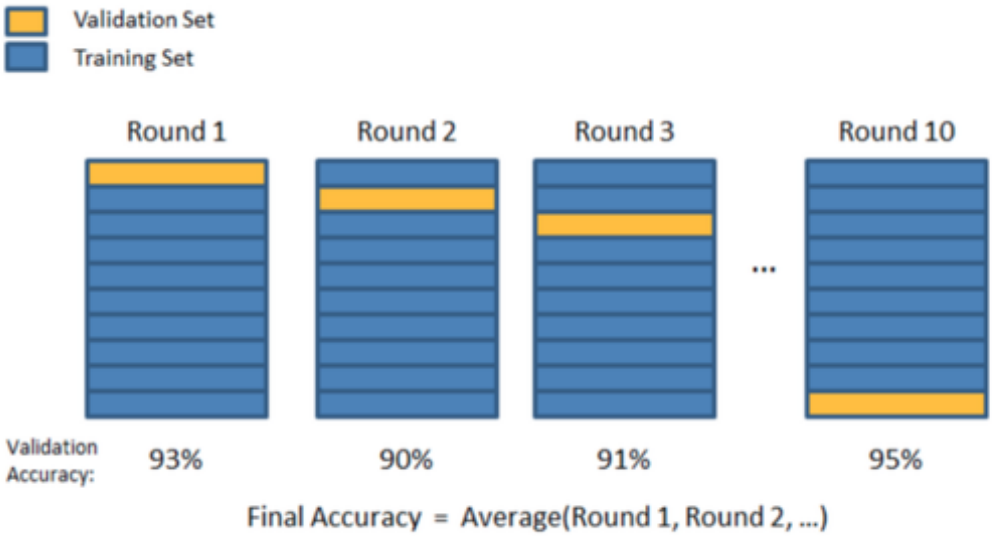


Figura 33: Método *K-Folds*: exemplo de validação cruzada com 10 grupos (*folds*). A métrica de avaliação, neste caso, a acurácia, é avaliada 10 vezes para cada grupo e então calculada a média. Fonte: (4)

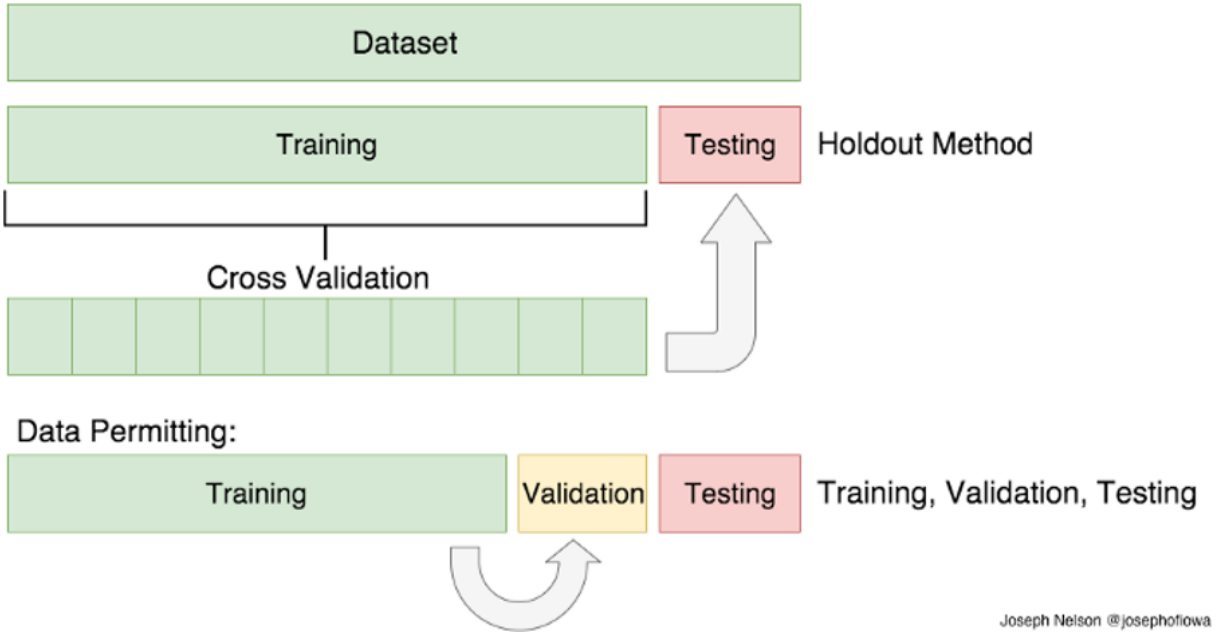


Figura 34: Esquema da divisão dos subconjuntos treino, teste e validação, pelo método *Cross-Validation*. Fonte: (4)

5.3.3.2 Inicialização dos modelos - *Random Forest* e *K-Nearest Neighbors*

A partir da biblioteca *SciKit Learn* (24), a implementação dos modelos é direta e convenientemente parametrizável, sendo necessária apenas a importação de cada uma das classes (*Random Forest* (25) e *K-Nearest Neighbors* (26)). A partir da importação já é possível treinar o modelo e avaliar suas respectivas performances, no entanto o grupo definiu como fundamental um estudo anterior de definição dos hiperparâmetros de cada modelo. Define-se hiperparâmetros como definições do modelo em si (não relacionado diretamente aos dados). Assim, é possível avaliar, para cada modelo, quais são os hiperparâmetros que melhor performam através de métodos de validação cruzada, especificamente o método *GridSearchCV* da biblioteca *SciKit Learn* (27). Para cada um dos modelos foram avaliados os seguintes hiperparâmetros:

1. ***K-Nearest Neighbors:***
   (a) Número de vizinhos (*n neighbors*);
   (b) Pesos (*weights*);
2. ***Random Forest:***
   (a) Número de árvores (*n estimators*);

O modelo ***K-Nearest Neighbors*** (26) performa a classificação avaliando uma distância, euclidiana nesta implementação, entre a amostra a ser avaliada e a base de dados. Feitos os cálculos da distância, o modelo segue a predição recuperando a classificação dos k vizinhos mais próximos e ponderando o peso de cada classificação na predição - a depender dos hiperparâmetros. Já o modelo ***Random Forest*** (25) funciona de maneira semelhante, porém instanciando diferentes árvores de decisão (28) e avaliando de maneira análoga a classificação resultante de cada árvore e ponderação de seus resultados.

5.3.4 SMOTE - Oversampling

Durante as observações sobre os resultados obtidos nos testes do algoritmo, notou-se que a métrica *recall* estava muito baixa (em torno de 30%), e para os objetivos desse projeto, essa métrica deveria ser o mais alto possível. A métrica de *recall* indica a razão entre a quantidade de resultados classificados como verdadeiros corretamente com o valor total de reais verdadeiros, ou seja, se existem 50 reais verdadeiros, e o algoritmo acerta a classificação de 20 deles, o *recall* vale 40%, independente dos valores verdadeiros classificados erroneamente.

Para melhor compreensão do conceito da métrica recall, imagine a IA que classifica resultados de um exame de câncer. Essa classificação idealmente alerta todos os reais casos de câncer, ainda que classifique reais falsos como verdadeiros. Nesse contexto, não identificar um verdadeiro, pode desviar o paciente do tratamento, diminuindo suas chances de sobrevivência. E nos casos que a IA retorne positivo, o médico responsável deve investigar com mais exames se o paciente realmente está com a doença, e assim aqueles falsos serão descartados com segurança e os pacientes que realmente precisam iniciar os devidos tratamentos serão orientados corretamente.

Para o nosso projeto, as análises que retornam verdadeiro são as empresas que são passíveis de investimento segundo a base histórica fornecida pelo investidor. Dessa maneira, quanto maior o *recall*, melhor é a taxa de identificação das empresas que possivelmente receberão investimentos, isto é, aumentamos a cobertura das oportunidades de investimento. No contexto deste projeto, classificar um real verdadeiro como falso é camuflar uma oportunidade de investimento.

O problema de um *recall* muito baixo pode ser causado pelo desbalanceamento entre as amostras negativas e positivas na base de dados, sendo que as amostras negativas prevalecem na base, gerando um modelo tendencioso para classificar como falso. No panorama de decisão de investimento, é razoável entender que haja maior quantidade

de resultados negativos a positivos. Para diminuir o desequilíbrio na base, é possível aumentar a quantidade de registros positivos, ou diminuir a quantidade dos negativos. Optou-se pelo uso da técnica de *oversampling* (aumento dos positivos), justificado pela preferência de bases com mais dados quando se trata da área de ciência de dados.

Para aplicar o *oversampling* no algoritmo, a biblioteca Imbalaced-Learn (29) em Python fornece o método SMOTE (30). Após aplicar esse tratamento nos dados, a métrica *recall* aumentou para valores acima de 70%. Outra consequência foi o aumento da taxa de reais falsos classificados como verdadeiros, mas analogamente ao contexto dos exames de câncer, essas classificações como verdadeiras devem ser analisadas com mais profundidade, e somente então o aporte será aprovado ou não. Apesar da necessidade da análise manual o sistema desenvolvido reduz a quantidade de empresas que terão que ser analisadas com mais detalhes.

5.3.5 Simulações de alteração na empresa analisada - sugestões de melhoria

Para as empresas que foram analisadas e não conseguiram aprovação para aportes (apenas para melhor compreensão, essa empresa será referenciada nessa seção do texto por "empresa negada"), o sistema automaticamente procura por empresas similares na base histórica do investidor que receberam aportes no passado. A definição de qual empresa semelhante a ser utilizada nas simulações para sugestão de melhoria é feita com base na distância cossenoidal (31) entre os registros das empresas, para melhor compreensão, esta empresa será referenciada por "empresa similar".

A escolha da metodologia pela distância cossenoidal (31) foi baseada em ferramentas de buscas que também usam do mesmo método internamente. O conceito se baseia na construção de vetores em que cada "dimensão" é definida pelos valores de cada parâmetro. Então cada empresa define um vetor, e os vetores mais próximos indicam as empresas mais semelhantes. Esta distância foi calculada pelo método *cdist* da biblioteca Scipy (32) em Python, com este método é possível calcular a distância entre dois registros por métodos diversos, como distância Euclidiana ou Manhattan, sendo configurado pelo parâmetro *metric* (terceiro passado na chamada da função). Para calcular a distância cossenoidal, basta passar o valor *'cosine'* para o parâmetro *metric*. As distâncias são retornadas pelo método, e ordenadas de maneira crescente.

Para a definição da "empresa similar", o registro relativo à menor distância é o primeiro a ser testado. Antes de começar as simulações de alteração, é necessário validar

que a empresa real verdadeira é classificada pelo modelo como verdadeira, apesar das distâncias terem sido calculadas apenas em relação a empresas que são real verdadeiras, o algoritmo não apresenta taxa de recall de 100%, e portanto pode classificar empresas real verdadeira como falsa, e nessas situações a empresa candidata a "empresa similar" não é interessante. Enquanto a empresa candidata não seja aprovada pelo modelo, a próxima empresa mais semelhantes será testada até que se encontre uma empresa classificada como verdadeira. Esta empresa candidata é finalmente definida como a "empresa similar".

Definida a empresa similar, o algoritmo passa a realizar as simulações. Essas simulações consistem em fazer alterações nos valores dos parâmetros da "empresa negada" e avaliá-la novamente, toda análise positiva é apresentada para o usuário (tanto o empreendedor como o investidor). Uma análise positiva que mude o tempo de mercado da empresa de 12 meses para 24 meses indiretamente indica que a empresa tende a aumentar suas chances de receber aportes daqui um ano, ou seja, a empresa deve se manter no mercado por mais um ano para que realmente seja vista como uma boa oportunidade de investimento.

As simulações em cima da "empresa negada" são feitas em duas etapas. A primeira etapa realiza a alteração de apenas um parâmetro por rodada, e tenta encontrar o valor mais otimizado para esse parâmetro. E na segunda etapa, são alterados dois parâmetros simultaneamente.

A simulação que altera apenas um parâmetro por rodada testa inicialmente se a troca pelo valor bruto da "empresa similar" é suficiente para aprovação. Caso isso não ocorra, as simulações com este parâmetro finalizam e passa para o parâmetro seguinte. Nos casos em que o valor bruto resulta em aprovação pelo modelo, o algoritmo busca pelo valor que exija a menor alteração em relação ao estado atual da "empresa negada". Para tal objetivo, a cada rodada de análise o algoritmo testa a "empresa negada" com o valor médio entre o último valor aprovado e o último não aprovado do parâmetro em teste, até que a diferença entre esses valores seja menor ou igual a 5% do maior valor entre o valor inicial da "empresa negada" e o valor da "empresa similar".

Existem parâmetros que geralmente tendem a melhorar o resultado da análise aumentando o valor, e outros diminuindo, como é o casos dos parâmetros de receita mensal e de gasto por produto, respectivamente. E a metodologia empregada na otimização dos valores independe do parâmetro, ela segue a direção definida pela "empresa similar", seja para um valor crescente ou descrescente.

Nas simulações com testes em dois parâmetros a lógica é similar, porém neste caso o

valor da “empresa negada” é simplesmente substituído pelo valor da “empresa similar”, sem tentativas de otimização dos valores, dado que a alteração do valor de um parâmetro pode influenciar na otimização do outro parâmetro.

5.4 Validação e testes

Nesta seção do trabalho é abordada a metodologia utilizada no processo de validação e testes, assim como os resultados e as alterações realizadas como consequências das análises dos testes.

5.4.1 Ausência de banco de dados disponível para testes

Para validar a ideia do projeto, o algoritmo deveria ser simulado e testado em cima de uma base de dados. A primeira tentativa de montar essa base de dados foi procurar por bases de dados públicas sobre empresas reais, como Crunchbase (12) e datasets públicos do Kaggle (13). Mas por se tratarem de dados sensíveis e confidenciais, o grupo precisou recorrer a outras soluções. A outra solução levantada foi sintetizar os dados para simular uma base de dados fictícia.

No processo de síntese de dados, o grupo usou de ferramentas comuns em processos de *Data Augmentation*, as bibliotecas Pandas e scikit learn.datasets fornecem muitas funcionalidades que podem ajudar na sintetização dos dados.

Se tratando de dados em grandes volumes, também é extremamente importante o tratamento dos dados correto. Este passo possibilita que a manipulação se mantenha coerente, e não estrague a análise dos dados. Os trabalhos com grande volumes de dados foram feitos em cima de Dataframes fornecidos pela biblioteca Pandas (33). Alguns tratamentos que foram necessários incluem manipulação de campos de datas, binários, e que contém rótulos, como “Indústria” e “Curso do(s) Fundador(es)”. Além do cuidado com os tipos de dados, outros parâmetros foram implementados com base em parâmetros originais passados no modelo da base histórica, por exemplo, o tempo de mercado da empresa é calculado em meses com base na data de fundação da empresa.

5.4.2 Testes iniciais

Com a base de dados disponível, foram empregados métodos da própria biblioteca Scikit Learn, como train_test_split (4) para realizar as etapas de validação e testes. É in-

teressante retomar os conceitos de validação e testes, e identificar as diferenças entre eles. Segundo Suniga (34), os dados disponíveis devem ser separados em 3 grupos, treinamento, validação e testes. O primiero conjunto é usado no treinamento, e a partir dele que se define um modelo. A segunda parte dos dados é utilizada para validação do modelo, que nas palavras de Suniga significa "comparação de diferentes modelos e hiperparâmetros", isto é, são os dados que serão utilizados para comparar modelos com diferentes configurações, e em cima dos resultados, definir a melhor combinação dos parâmetros para o processo de modelagem. E por fim, o conjunto de treino é utilizado ao final do processo, para verificar a acurácia do modelo proposto, para então provar que o modelos funciona.

Passado os processos de validação e testes feitos pelo algoritmo, o grupo se empenhou para reportar métricas relevantes sobre o algoritmo. Para tal ovjetivo, a biblioteca Scikit Learn também disponibiliza métodos que auxiliam na análise dos modelos gerados, provendo bibliotecas que retornam a matriz de confusão e relatórios com as principais métricas de classificação, as bibliotecas mencionadas são confusion_matrix (35) e classification_report (36).

5.4.3 *Overfitting* e baixa taxa de *recall*

O Overfitting é um problema muito discutido na área de aprendizado de máquina por ser muito comum nos algoritmos. No projeto em discussão, o grupo também precisou lidar com esse tipo de inconsistência, e focado principalmente nos casos negativos. A acurácia do algoritmo estava extremamente alta, indicando possível overfitting no treinamento dos dados, e a comprovação se deu ao analisar acurácia do modelo nos conjuntos de testes.

Como soluções, o grupo optou por usar o modelo Random Forest, além do modelo K-NN, e reduzir o número de parâmetros nas análises. A escolha pelo modelo Random Forest foi sugerida pelos orientadores, além da escolha dos métodos, a escolha dos parâmetros dos modelos é de extrema relevância, definindo valores coerentes problema de overfitting pode ser contornado. Para a otimização dos parâmetros de ambos os modelos, a biblioteca Scikit Learn oferece o método GridSearchCV (27) para testes de valores de parâmetros para a base de dados analisada.

Como comentado, outra solução implementada foi a redução de parâmetros utilizados na modelagem da tese. Dos 19 parâmetros originais pedidos pelo modelo de arquivo para a base histórica, são usados 5 deles. A escolha dos parâmetros é feita automaticamente pelo algoritmo, elecando as 5 colunas mais correlacionadas com a variável-alvo "Investiu".

Ainda se tratando sobre overfitting, o desbalanceamento entre registros positivos e negativos também causou uma taxa muito baixa de recall. Como detalhado na seção prévia de desenvolvimento do algoritmo, o recall foi tratado com técnicas de oversampling através do método SMOTE (30) fornecido pela biblioteca Scikit Learn. O oversampling deixa a base de dados mais equilibrada entre dados positivos e negativos, diminuindo o problema de overfitting de casos negativos, e portanto é capaz de cobrir melhor as oportunidades de investimentos que chegam ao investidor.

PARTE IV

RESULTADOS

6 ANÁLISE DOS DADOS SINTETIZADOS

O grupo optou pela abordagem de sintetizar os dados necessários para treinamento dos modelos, por dois principais motivos: garantir a coerência com a estrutura de dados montada e preservar a confidencialidade das informações necessárias - pois cada fundo trata seu histórico de análises com sigilo. Assim, o tema de sintetização de dados estruturados fora um estudo à parte e complementar ao projeto principal, mas não menos importante, de modo que fora escolhido como estratégia para contornar o problema de disponibilidade dos dados.

Inicialmente, fora criada manualmente uma base pequena de apenas 8 amostras, isto é, 8 empresas de acordo com as variáveis definidas. A partir disso, o grupo seguiu a metodologia definida e montou o *Data-Science Pipeline* de uma maneira parametrizada e apta a receber um conjunto maior de dados. Ao final dessa primeira rodada de treinamento dos modelos, fora confirmado que era um universo de amostras muito pequeno e pouco representativo. Tal constatação ficou evidente, principalmente, na observação das dimensões dos conjuntos de treino, teste e validação - este último recebendo apenas duas amostras.

Desse modo, o grupo criou uma base totalmente sintética a partir da função *make regression* (19) da biblioteca *scikit learn*. A priori, a base conteria 1000 amostras com valores originalmente aleatórios, e para serem tratados com os métodos *MinMaxScaler* (20) e *Normalize* (37), ambos da biblioteca *scikit learn*, para representarem a base original de 8 amostras. Essa estratégia melhorou a questão da dimensão dos conjuntos, permitindo que os testes fossem mais conclusivos e que as métricas (especialmente, acurácia e *recall*) de avaliação dos modelos tivessem valores coerentes (não nulos). No entanto, nesta segunda *sprint* fora constatada também a necessidade de um tratamento especial para a variável-alvo **"Investiu"** no conjunto sintético. Isso se confirmou, ao analisar que nenhuma das variáveis explicativas tinha correlação expressiva com a variável-alvo: através da matriz de correlação, notava-se que de modo geral eram quase nulas - o que é esperado, devido ao método de geração aleatório empregado. Mais uma vez, isso se refletiu na performance dos

modelos, obtendo uma acurácia em torno de 50%, o que também é esperado: se tratando de um problema de classificação com dois rótulos ("aprovar" ou "reprovar").

Finalmente, a conclusão dessa análise é de que a variável-alvo necessitava de definições especiais e relacionadas ao contexto do problema. Desse modo, após o tratamento e conversão dos números aleatórios originais das variáveis explicativas para o contexto e tipo de variáveis necessários, o grupou adotou rotinas para: eliminar amostras incoerentes (por exemplo, data de submissão no processo de análise anterior à data de fundação); e avaliar amostras que poderiam ser reprovadas. A seguir, as regras adotadas para cada um dos tipos de regras:

1. Eliminação de amostras irreais:
   (a) Amostras cujas datas não respeitavam a ordem cronológica esperada: data de fundação ¡ data de submissão ¡ data de decisão;
2. Classificação lógica da variável alvo (atribuição automática de **reprovações**):
   (a) Empresas recém-fundadas (menos de um ano de operação);
   (b) Decisões rápidas (data de submissão próxima à data de decisão): subentende-se uma eliminação rápida;
   (c) Empresas com número de funcionários acima da média da base (acima do 2º quartil) e que não geravam receita;
   (d) Empresas com custo de aquisição de cliente (CAC) acima da média da base (acima do 2º quartil) e que não geravam receita;
   (e) Empresas com poucos clientes (abaixo do 2º quartil), CAC alto (acima do 2º quartil) e clientes não recorrentes;
   (f) Empresas do ramo de tecnologia (Tech), sem produto próprio e cujo fundador não era da área (engenharia ou computação).

A partir dessas regras, o conjunto de 1000 amostras fora reduzido para 597 amostras reais, das quais 267 receberam a reprovação automática. O grupo avaliou tanto a redução de amostras quanto a classificação automática como benéficas para o modelo, uma vez que traz uma lógica a ser identificada pelos modelos preditivos mais adiante. No entanto, restaram ainda 330 amostras coerentes que precisavam ser avaliadas. Então o grupo fez um esforço de avaliação manual de todas as amostras restantes e decidiu se a empresa teria sido aprovada para receber investimentos, baseado nos critérios de outras startups

e pesquisa anterior sobre ciclo de vida das empresas. O resultado final foram que das 597 amostras identificadas como reais, 468 foram classificadas como "Não investir" e 129 como "Investir". Na figura 35 é possível observar os impactos de cada uma das etapas na organização dos dados sintéticos. Detalha-se ainda os impactos da ampliação de dados implementada em cada uma das variáveis, ao se comparar o conjunto antes das rotinas, tabela 3, e depois, tabela 4.

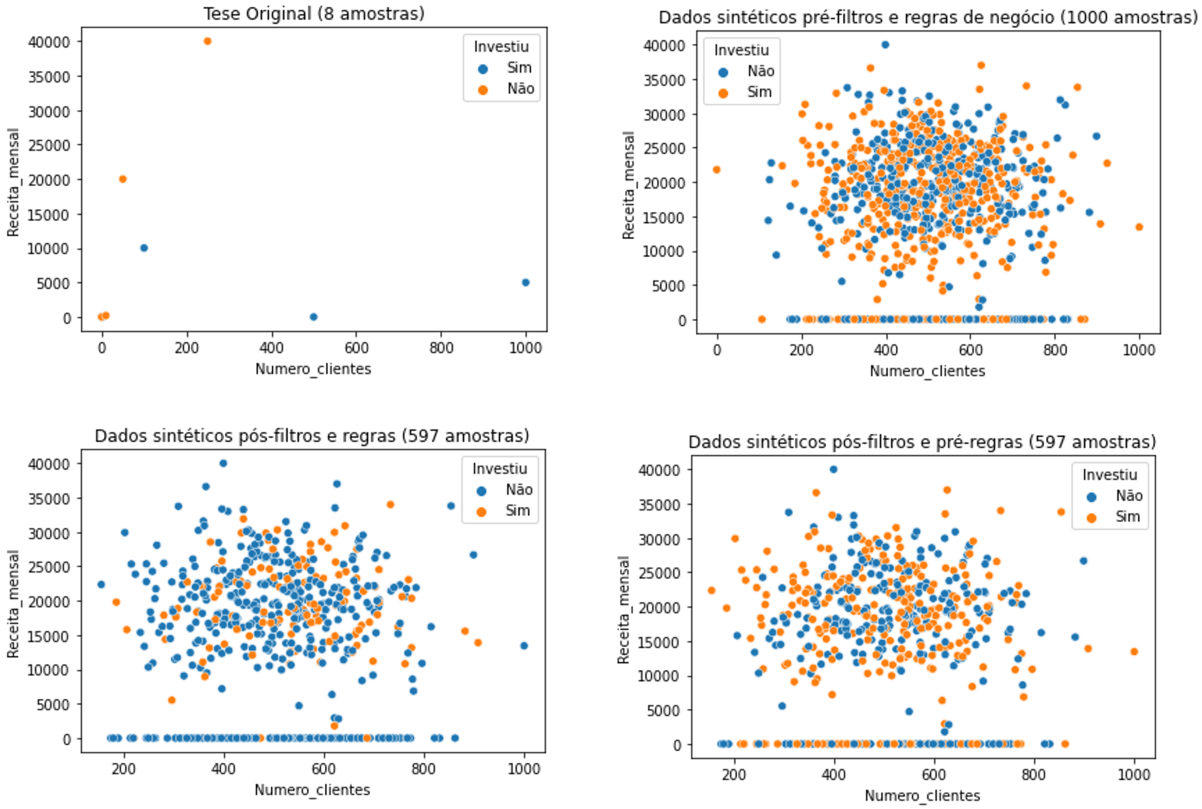


Figura 35: Gráficos de dispersão representando o universo de amostras a cada etapa do processo de *Data Augmentation*. Fonte: produzida pelos autores.

Métrica	**Resultado**
Tamanho da conjunto	8
Variáveis sintetizadas pela rotina	-
Nome da empresa	8 não nulos
Data da submissão	8 não nulos entre: 10/jan/2022 e 08/dez/2022
Data de fundação	8 não nulos entre: 02/jan/2021 e 09/jan/2022
Quantidade de funcionários	8 não nulos média 43, mínimo 5 e máximo 132
Indústria	8 não nulos Finanças (2), Logística (1), Sustentável (1), Tech (4)
Produto próprio?	8 não nulos Não (4) e Sim (4)
Gerando receita?	8 não nulos Não (3) e Sim (5)
Receita mensal (R$/mês)	8 não nulos média R$ 9.400,00, mínima R$ 0 e máxima R$ 40.000,00
Número de clientes	8 não nulos média 239, mínimo 0 e máximo 1000
Clientes recorrentes?	8 não nulos Não (4) e Sim (4)
CAC histórico (R$)	8 não nulos média R$ 573,75, mínimo R$ 200 e máximo R$ 1000
Curso Founder	8 não nulos Administração (1), Ciência da Computação (1), Economia (1), Engenharia (4), Química (1)
Data Decisão	8 não nulos entre: 19/abr/2022 e 10/dez/2022
Investiu?	8 não nulos Não (4) e Sim (4)

Tabela 3: Histórico de decisões submetido (conjunto inicial). Fonte: produzida pelos autores.

Métrica	Resultado
Tamanho da conjunto	597
Variáveis sintetizadas pela rotina	14
Nome da empresa	597 não nulos
Data da submissão	597 não nulos entre: 01/jan/2022 e 05/dez/2022
Data de fundação	597 não nulos entre: 02/jan/2021 e 23/jul/2022
Quantidade de funcionários	597 não nulos médio 68, mínimo 5 e máximo 116
Indústria	597 não nulos Finanças (313), Logística (19), Sustentável (11), Tech (254)
Produto próprio?	597 não nulos Não (263) e Sim (334)
Gerando receita?	597 não nulos Não (154) e Sim (443)
Receita mensal (R$/mês)	597 não nulos média R$ 14.756,43, mínimo R$ 0 e máxima R$ 40.000,00
Número de clientes	597 não nulos médio 511, mínimo 155 e máximo 1000
Clientes recorrentes?	597 não nulos Não (292) e Sim (305)
CAC histórico (R$)	597 não nulos média R$ 577,01, mínimo R$ 228 e máximo R$ 1000
Curso Founder	597 não nulos Administração (350), Ciência da Computação (145), Economia (87), Engenharia (13), Química (2)
Data Decisão	597 não nulos entre: 07/fev/2022 e 10/dez/2022
Investiu?	597 não nulos Não (468) e Sim (129)

Tabela 4: Histórico de decisões submetido (conjunto inicial). Fonte: produzida pelos autores.

O grupo avaliou que o resultado observado estava em linha com o esperado, pois num contexto de decisão de investimento, de fato é a minoria das aplicantes que recebem sinal positivo para a operação de investimento. Além disso, as métricas de acurácia e *recall* dos modelos também subiram consideravelmente, como relatado no próximo capítulo. Adiante, no capítulo de trabalhos futuros, se faz uma breve discussão sobre essas regras de negócio. Comparando cada conjunto de dados, destacam-se algumas limitações das técnicas de ampliação de dados. Primeiramente, a necessidade de reduzir para amostras coerentes com a aplicação (como descrito na seção de desenvolvimento), isto é, filtrar novamente os dados para garantir que são simulações razoáveis e factíveis. Por fim, o fato de que os limites (tanto mínimo quanto máximo) impostos pelo conjunto original serão mantidos nos conjuntos consecutivos, como se observa nas tabelas 3 e 4.

7 ANÁLISE DOS ALGORITMOS E MÉTRICAS

Para os usuários, o resultado final serão as devolutivas dos modelos, ilustradas nas figuras 16 e 17 supracitadas - aprovação ou reprovação, respectivamente. A partir dessas devolutivas, o grupo pôde também fazer uma análise sobre a qualidade do algoritmo desenvolvido e se o objetivo fora alcançado ou não. Destaca-se que a ferramenta almeja ser um suporte à decisão de investimento não configurando a decisão em si, mas sim fornecendo aos usuários um *feedback* do que pode ser melhorado nos casos de reprovação.

A avaliação do grupo fora feita a partir do conjunto resultante dos métodos de *Data Augmentation* supracitados. Conforme mencionado no capítulo anterior, a seleção de variáveis a serem utilizadas pelo método de validação cruzada é fundamental para execução adequada do algoritmo. Sem a seleção de variáveis, o grupo observou uma alta tendência ao *over-fitting*, com acurácia acima de 90% no conjunto de validação e em torno de 70% no conjunto de treino. Além disso, a seleção de variáveis é entendida como importante para o escopo do problema estudado, afinal diferentes investidores terão diferentes variáveis a considerar e mesmo que optem por coletar muitas informações, apenas algumas realmente são determinantes para a decisão. O método *RFECV* (23) da biblioteca *SciKitLearn* mistura as variáveis escolhidas para determinar: o número ótimo de variáveis; quais variáveis irão compor o modelo. A figura 36 demonstra o resultado dessa verificação, considerando cinco instâncias do classificador *Random Forest*.

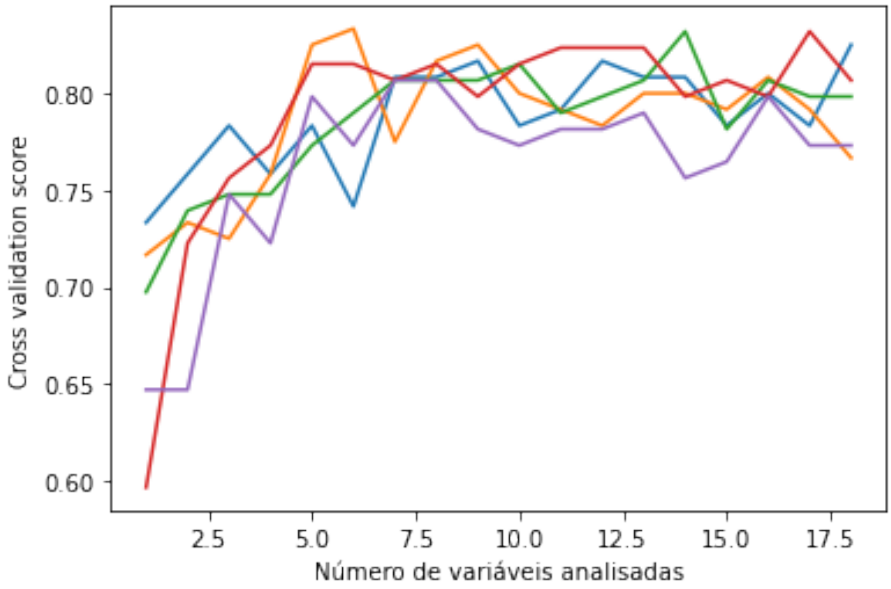


Figura 36: Performance de cada instância de classificador ao testar diferentes combinações de variáveis. Fonte: produzida pelos autores.

O grupo avalia que esta etapa de seleção de variáveis tem um amplo horizonte para melhorias, uma vez que fora implementada nesta versão da ferramenta uma combinação de diferentes métodos (análise de correlação final e *grid-search*) para selecionar algumas variáveis. Como o objetivo do trabalho fora definido em criar uma estrutura de análise, o grupo deu prioridade à comparação de diferentes modelos e para isso tomou a decisão de combinar as estratégias supracitadas para seleção final. Ao final da seleção pelo *grid-search*, o grupo restringiu ainda mais, dentro das variáveis selecionadas, ao considerar apenas as cinco variáveis que apresentaram maior correlação com a variável alvo. Essa opção de restringir ainda mais, fora justificada a partir dos cálculos de acurácia que ainda demonstravam forte tendência ao *over-fitting*. A quantidade definida de parâmetros (cinco) foi feita a partir do tamanho da base de aprendizado (base histórica de investimentos), portanto esta quantidade poderia ser flexível, dependendo somente da quantidade de registros disponíveis.

Quanto à performance dos modelos em si, é possível avaliá-los através do método *Classification-Report* (36) nativo da biblioteca *SciKit-Learn*. As tabelas 5 e 6 se referem justamente ao resultado deste método para os modelos *K-Nearest Neighbors* e *Random Forest*. Adicionalmente, o modelo *Random Forest* possui um método nativo para cálculo da importância das variáveis, *feature-importance* (38). Tal método pode ser utilizado para visualizar graficamente, figura 37, a importância de cada variável no modelo (para o modelo K-NN, não há uma implementação nativa de análise semelhante).

	precision	recall	f1-score	support
False	0.79	0.97	0.87	117
True	0.00	0.00	0.00	31
accuracy	-	-	0.77	148
macro avg	0.39	0.49	0.44	148
weighted avg	0.62	0.77	0.69	148

Tabela 5: *Classification Report* do modelo K-NN para os dados sintéticos. Fonte: Produzida pelos autores

	precision	recall	f1-score	support
False	0.81	0.97	0.88	117
True	0.56	0.16	0.25	31
accuracy	-	-	0.80	148
macro avg	0.68	0.56	0.57	148
weighted avg	0.76	0.80	0.75	148

Tabela 6: *Classification Report* do modelo *Random Forest* para os dados sintéticos. Fonte: Produzida pelos autores

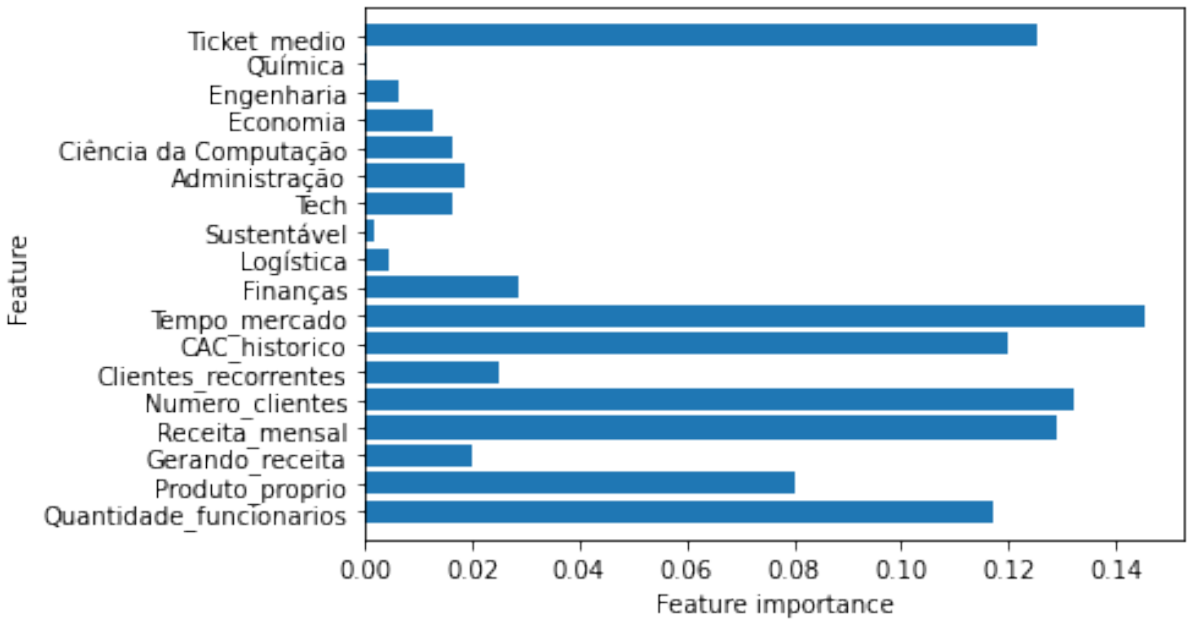


Figura 37: Gráfico representando a importância de cada *feature*, refletida por um número de 0 a 1 em que: 0 significa "não usada" e 1 "perfeitamente prevê a resposta". Destaca-se que a importância de todas as *features* do modelo somam sempre 1. Fonte: Produzida pelos autores.

A conclusão do grupo é de que a estrutura do modelo e da sequência lógica das etapas do *Data Science Pipeline* estão coerentes e permitem uma avaliação de diferentes modelos e métricas, em linha com o objetivo de criar uma ferramenta parametrizável. No entanto, as métricas de avaliação indicam uma baixa qualidade dos dados recebidos (conjunto sintético), sendo que isso fora presente ao longo de todo o desenvolvimento do trabalho. O grupo avalia que os trabalhos futuros seriam a condução de testes com bases reais e que tenham alguma lógica intrínseca para justamente avaliar uma melhora nos índices supracitados. Ainda assim, o sistema fora montado de maneira a evidenciar essas situações e permitir análises como essa e o seu refino em projetos futuros.

8 LANDING PAGE E VÍDEO DE DEMONSTRAÇÃO

Foi criado também uma página pelo serviço Google Sites para divulgação do projeto (39). O intuito da página é centralizar as informações do projeto de uma maneira sintética e próxima aos usuários interessados. Além de referências aos relatórios e repositórios, na página também fora incluído um vídeo de demonstração da ferramenta (40), de acordo com a descrição neste documento. O vídeo explica de maneira sintética ambos os fluxos dos usuários esperados (investidores e empreendedores), destacando o que está sendo realizado por trás da interface de comunicação com o usuário e explicando brevemente as lógicas detalhadas neste relatório.

PARTE V

CONCLUSÃO

9 IMPLEMENTAÇÃO DA FERRAMENTA NUM FUNDO REAL DE INVESTIMENTOS

Para um usuário-real conseguir usar a ferramenta, é esperado um registro prévio de todas as empresas que já analisou e respectivas decisões. A ferramenta disponibiliza um *template* desse histórico a ser enviado, de modo que basta ao investidor preenchê-lo de acordo com sua tese. Numa situação em que as variáveis já implementadas contemplam as análises do investidor, não há a necessidade de fazer nenhuma alteração no sistema interno, espera-se que o histórico seja suficientemente significativo em termos de número de amostras, e que possa ser feita a desativação do módulo de *Data augmentation* (apesar de ainda estar disponível, em caso de uma eventual necessidade).

Por outro lado, numa situação em que as variáveis previstas no sistema não contemplem as informações necessárias para a avaliação, o grupo poderá incluir novas colunas na estrutura dos algoritmos. Devido à construção parametrizada e separada por etapas, de acordo com o *Data Science Pipeline*, cada nova variável deverá ser analisada independentemente e repetir as etapas de: limpeza de dados e engenharia de variáveis. Ou seja, será necessário avaliar e pré-definir os formatos das novas variáveis para o correto funcionamento do modelo. Com isso concluído, basta seguir o *pipeline* conforme descrito acima, em que será analisada a importância de cada variável e será feita a seleção. O grupo entende que essa é uma responsabilidade constante e missão perene na manutenção da ferramenta, uma vez que a mesma deve estar em total conformidade com as tendências do mercado e análises necessárias. Outra solução para o problema em discussão seria o uso da ontologia montada para poder adaptar o conteúdo entre bases com diferentes parâmetros, assim investidor e fundador podem apresentar seus dados da maneira que lhes servir, e o algoritmo seria capaz de interpretá-los corretamente.

Para a avaliação da necessidade ou não dessas mudanças, recomenda-se que o grupo acompanhe o usuário investidor numa jornada de *onboarding*, na qual fará a introdução e explicação do propósito da ferramenta. Destaca-se ainda que, apesar de terem sido implementados dois fluxos de usuários, é esperado que o primeiro usuário a adotar a fer-

ramenta seja o investidor, uma vez que suas teses devem estar disponíveis para simulações dos empreendedores.

Para enriquecer esta discussão e validar as percepções levantadas, o grupo se organizou para entrevistar possíveis usuários do sistema, através do co-orientador Victor, os alunos conversaram com Guilherme Passos e Renan Oliveira do *family office* Anima Investimentos. O primeiro contato ocorreu no começo do ano, para apresentação do problema e breve discusão da solução proposta. Dado o interesse dos gestores, e a confirmação de uma dor real, o grupo prosseguiu com os estudos. Ao final do desenvolvimento, o grupo se encontrou novamente com o mesmo grupo de gestores para apresentar a ferramenta implementada e receber *feedbacks*. A discussão foi bastante positiva e o produto bem recebido pelos entrevistados. Um dos convidados trouxe comentários sobre a relação entre rentabilidade no passado e a garantia ou não de rentabilidade no futuro. Além de melhorias que já estavam mapeadas, como inserção de variável *booleana* de sucesso do investimento e aprendizado constante do algoritmo (ambos planejados para trabalhos futuros). Além disso, foram feitas sugestões sobre a ponderação entre o peso de cada amostra na classificação do modelo de acordo com a recência da mesma amostra. Ao mesmo tempo, o convidado mencionou que alguns fundos podem ter a preferência por pesos iguais em todas as análises, guiando para uma solução ainda mais personalizada para cada investidor. Na mesma linha de customização, também foi sugerido que o usuário-empreendedor possa refinar suas buscas, como filtrar por fundos com mais tempo de experiência e existência no mercado.

10 IMPLEMENTAÇÃO DA FERRAMENTA EM OUTROS CONTEXTOS DE INVESTIMENTOS

Conforme descrito nas seções iniciais, o presente trabalho se aprofunda no cenário de investimentos em *startups*, uma vez que fora analisado o ciclo de vida dessas empresas e as tendências atuais para definir as variáveis a serem consideradas numa avaliação preliminar de sucesso da empresa ou não. No entanto, a proposta da ferramenta e a metodologia escolhida é de agregar valor em diferentes situações em que é necessária uma triagem de um grande volume de candidatos para receber investimentos. Destaca-se aqui editais de apoio e fomento à pesquisa ou projetos universitários, cada vez mais presentes e desempenhando um papel importantíssimo no crescimento da ciência e empreendedorismo brasileiros.

Dessa forma, visando essa possibilidade futura de expansão de uso, o grupo construiu a ferramenta totalmente personalizável. Para a adaptação de cenários, será imprescindível um entendimento a fundo do caso e definição das variáveis a serem consideradas, além de também ser vital a disponibilidade de uma base de dados confiável (ou recorrer aos métodos de *data augmentation*, com as ressalvas da seção anterior). Assim, a partir da definição de quais variáveis serão utilizadas, também será necessário refazer as análises das variáveis e formatação dos dados, de modo a garantir que essas novas variáveis sejam corretamente interpretadas pelos modelos. Uma vez feitas essas adaptações, os modelos e filtragens funcionarão de modo semelhante, já que as rotinas de seleção de variáveis não foram feitas de maneira engessada e específica, mas sim parametrizada e preparada para receber novos contextos.

11 AVALIAÇÃO DO GRUPO QUANTO AO PROJETO E CURSO DE ENGENHARIA

O grupo entende que o projeto fora direcionado de acordo com os princípios de engenharia, e em especial, à estrutura da Escola Politécnica da USP. Os conceitos estudados e exercitados em múltiplas disciplinas ao longo do curso, como, por exemplo, as matérias básicas de matemática (cálculo e álgebra linear) e específicas da engenharia elétrica e computação, foram fundamentais para o desenvolvimento da capacidade de abstração e detalhamento de um plano estratégico de resolução de problemas. As técnicas de aprendizado de máquina e modelos preditivos estão intrinsecamente relacionadas às bases matemáticas vistas nas disciplinas de álgebra linear, cálculo e cálculo numérico. Destaca-se também a necessidade de ter uma base sólida em estatística e probabilidade, para avaliar tanto os modelos e sua performance quanto os dados sintetizados.

Sobre as tecnologias e metodologias utilizadas, o grupo avalia que fora vital o entendimento prévio dos algoritmos avaliados e de aprendizado de máquina - estudados nas disciplinas de algoritmos e intelig^encia artificial. O grupo também exercitou princípios modernos de desenvolvimento de softwares ao procurar implementações já consolidadas e construir o projeto com base nelas para alcançar o objetivo final desejado. Destaca-se neste sentido a utilização das bibliotecas Pandas (41) e SciKit-Learn (24) e utilização de métodos de autoria própria para completar o sistema final. Outras matérias do módulo vermelho foram bastante relembradas durante o processo, conceitos como diagramas de caso de uso e estruturação de um banco de dados foi revisitados na página da disciplina PCS3413 - Engenharia de Software e Banco de Dados (42); metodologias das disciplinas de laboratório de Engenharia de Software (I e II) foram adaptadas ao contexto e ao grupo para serem colocadas em prática, assim como os aprendizados provenientes do projeto desenvolvido em curso.

As *soft skills* complementam o quadro de características desejadas em um aluno. Durante o desenvolvimento do projeto, o grupo reforçou habilidades como comprometimento, gestão de tempo, resiliência, divisão de tarefas e trabalho em grupo. As habilidades men-

cionadas foram constantemente exigidas ao longo do curso, formando bons cidadãos e profissionais.

Finalmente, o grupo reconhece o posicionamento da Escola Politécnica da USP como formadora de empreendedores e a crescente tendência no mercado de empresas com modelos disruptivos proporcionando novas tecnologias e serviços. Conforme mencionado anteriormente, fora através das disciplinas de empreendedorismo que o grupo se motivou a estudar o tema a fundo e definiu o objetivo deste trabalho, de modo a propor alguma melhoria na experiência dos agentes envolvidos numa operação de investimento e aliviar as dores mapeadas inicialmente. Entende-se que a ferramenta desenvolvida tem um potencial promissor caso siga as propostas de incrementos levantadas no capítulo seguinte. O grupo vê a versão atual da ferramenta como uma espécie de *"MVP" (Minimum Viable Product)* e propôs um *roadmap* do que seriam os próximos passos para implementações futuras e consolidação da ferramenta num serviço com valor no mercado.

12 TRABALHOS FUTUROS

Em virtude do caráter de demonstração e objetivo de alcançar a validação dos principais conceitos na ferramenta desenvolvida, o grupo optou por focar no desenvolvimento das funcionalidades essenciais do sistema. Apesar de ter muito espaço para melhorias, fora desenvolvidos os pontos de maior valor para o usuário, podendo ser considerado como um *"MVP" (Minimum Viable Product).*

Para um produto final, o grupo entende que devem ser desenvolvidos os fluxos de *log-in's*, isto é, verificação de usuários já cadastrados e recuperação de suas informações. Esse fluxo já fora iniciado, uma vez que o cadastro e armazenamento das informações relativas ao mesmo está funcionando de maneira apropriada, restando então a implementação da verificação de identidade dos usuários (exigir senha, conferir se está correta e principalmente armazaná-la de maneira segura e encriptografada). Também há a necessidade de disponibilizar a ferramenta para acesso remoto pelos usuários, o grupo explorou para isso o serviço Streamlit Cloud (43), mas não avançou na sua implementação nesta fase de demonstração, pois priorizou a agilidade da demonstração e proposta de valor do sistema.

Ainda no tema de usabilidade do sistema enquanto produto, o grupo enxerga possibilidade de melhoria nos fluxos do investidor e empreendedor, através de uma interface mais interativa e personalização ainda maior das rotinas. Por exemplo, o sistema poderá verificar o tamanho da amostra de tese submetida pelo investidor e solicitar intervenção do usuário para prosseguir com a rotina de *data augmentation* - em especial na seleção de regras de negócios coerentes com a visão do investidor, e na avaliação de algumas empresas sintéticas. Destaca-se também que a própria rotina de *data augmentation* está sujeita a revisão, uma vez que for conduzido um processo de onboarding de investidor e avaliada a precisão desses métodos.

A construção da ferramenta fora idealizada de modo que o resultado final fosse objeto de novas iterações e incrementos de maneira fácil. Assim, deseja-se para os projetos futuros, implementar a possibilidade de adicionar tanto novas variáveis explicativas (seguindo o modelo da ontologia desenvolvido pelo grupo) quanto novos modelos preditivos.

A primeira, terá como objetivo ampliar as opções de variáveis a serem analisadas pelos investidores e contemplar mais teses de investimentos. Já os novos modelos, permitirão uma avaliação melhor embasada sobre quais são os métodos mais eficazes, para cada caso. Essas adições, poderão ser feitas de maneira descomplicada no corpo do algoritmo, nas respectivas seções - aquisição e preparação de dados; e treinamento dos modelos de classificação.

Para o algoritmo de modelagem, como já mencionado ao longo do texto, também podemos refinar a definição das variáveis utilizadas nas análises, se baseando em métodos disponíveis, como RFECV e *grid search*, e na quantidade apropriada para cada base de dados. Os parâmetros dos modelos deveriam ser definidos da maneira mais eficiente, otimizando os resultados e não levando muito tempo na definição.

Finalmente, o grupo pondera que o destino final da ferramenta dependerá da condução dos próximos passos listados acima e das parcerias comerciais conquistadas para testes com usuários reais e validação do *product market fit* (44). No entanto, destaca-se uma visão de posicionamento de produto coerente com o objetivo do grupo de mitigar as dores mapeadas na introdução deste trabalho: um *Customer Relationship Management (CRM)* (45) e canal de comunicação (46). O sistema desenvolvido se assemelha a um CRM sob a ótica do fluxo do investidor, uma vez que propõe como valor a gestão dos *leads* e priorização daqueles que têm maior aderência à tese de investimento. Ao mesmo tempo, para os empreendedores fora mapeado que não é produtivo a exigência de ter que fazer o mesmo cadastro em diferentes plataformas, assim a ferramenta, se posicionando como um canal independente e ao adicionar novas funcionalidades de contato entre os usuários, terá a oportunidade de mitigar essa dor e possibilitar ao empreendedor reutilizar seu cadastro para múltiplos investidores.

REFERÊNCIAS

1 MOORE, G. A. *Crossing the Chasm, Marketing and Selling High-Tech Products to Mainstream Customer*. New York: HarperCollins Publishers, 1999.

2 ENDEAVOR. *O Ciclo de Vida de uma Empresa de Sucesso*. 2022. Disponível em: ⟨https://endeavor.org.br/tomada-de-decisao/o-ciclo-de-vida-de-uma-empresa-de-sucesso/⟩. Acesso em: 5 de nov. de 2022.

3 DISTRITO. *Rodada de investimento: entenda como é o seu funcionamento*. 2022. Disponível em: ⟨https://distrito.me/blog/rodada-investimento-seed-series-a/⟩. Acesso em: 5 de nov. de 2022.

4 GENERAL ASSEMBLY, DC. *Train/Test Split and Cross Validation in Python*. 2022. Divisão dos conjuntos de teste, treino e validação no Python. Disponível em: ⟨https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6⟩. Acesso em: 12 de set. de 2022.

5 CHEROMBIM, A. P. M. S. et al. Capital de risco no brasil: a atuação do fundo de capital semente criatec. *A Revista Acadêmica da FACE*, 2011.

6 CNN BRASIL. *Investimentos em startups do Brasil somou US 9,4 bi em 2021, aponta levantamento*. 2022. Disponível em: ⟨https://www.cnnbrasil.com.br/business/investimento-em-startups-do-brasil-somou-u-94-bi-em-2021-aponta-levantamento/⟩. Acesso em: 10 de ago. de 2022.

7 CBINSIGHTS. *The State of Venture: Q3 2022 Global Report*. 2022. Disponível em: ⟨https://www.cbinsights.com/research-state-of-venture/⟩. Acesso em: 9 de dez. de 2022.

8 SAHLMAN, W. A. The structure and governance of venture-capital organizations. *Journal of Financial Economics*, v. 27, n. 2, p. 473–521, 1990. ISSN 0304-405X. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/0304405X90900658⟩.

9 FORBES MONEY. *Qual a diferença entre private equity e venture capital*. 2022. Disponível em: ⟨https://forbes.com.br/forbes-money/2022/08/qual-a-diferenca-entre-private-equity-e-venture-capital/⟩. Acesso em: 15 de out. de 2022.

10 UNIVERSIDADE DE SãO PAULO. *Disciplina: PCS3529 - Criação e Administração de Empresas de Computação*. 2022. Disponível em: ⟨https://uspdigital.usp.br/jupiterweb/obterDisciplina?sgldis=PCS3529⟩. Acesso em: 10 de ago. de 2022.

11 INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING (ICSE). *The Art and Practice of Data Science Pipelines*. 2022. Data Science Pipeline. Disponível em: ⟨https://dl.acm.org/doi/abs/10.1145/3510003.3510057⟩. Acesso em: 12 de set. de 2022.

12 CRUNCHBASE INC. *Crunchbase*. 2022. Página inicial da plataforma Crunchbase. Disponível em: ⟨https://www.crunchbase.com/⟩. Acesso em: 11 de dez. de 2022.

13 KAGGLE INC. *Kaggle*. 2022. Página inicial da plataforma Kaggle. Disponível em: ⟨https://www.kaggle.com/⟩. Acesso em: 11 de dez. de 2022.

14 CHURCHILL, N.; LEWIS, V. The five stages of small business growth. *Harvard Business Review*, 1986.

15 COSTA, A. H. R.; HRUSCHKA, E. R. *PCS3438 - Inteligência Artificial (2021)*. 2021. Página web da disciplina PCS3438. Disponível em: ⟨https://edisciplinas.usp.br/course/view.php?id=91586⟩. Acesso em: 11 de dez. de 2022.

16 SNOWFLAKE INC. *Streamlit*. 2022. Biblioteca Streamlit. Disponível em: ⟨https://streamlit.io/⟩. Acesso em: 09 de nov. de 2022.

17 SNOWFLAKE INC. *Streamlit Forms*. 2022. Formulários Streamlit. Disponível em: ⟨https://docs.streamlit.io/library/api-reference/control-flow/st.form⟩. Acesso em: 09 de nov. de 2022.

18 MONGODB, INC. *Field Update Operators*. 2022. Documentação sobre os campos disponibilizados para operações de atualização no MongoDB. Disponível em: ⟨https://www.mongodb.com/docs/manual/reference/operator/update-field/⟩. Acesso em: 11 de dez. de 2022.

19 SCIKIT-LEARN DEVELOPERS. *sklearn.datasets.make_regression*. 2022. Documentação do método make_regression da biblioteca SKLearn em Python. Disponível em: ⟨https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_regression.html⟩. Acesso em: 04 de out. de 2022.

20 SCIKIT-LEARN DEVELOPERS. *sklearn.preprocessing.MinMaxScaler*. 2022. Documentação do método MinMaxScaler da biblioteca SKLearn em Python. Disponível em: ⟨https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html⟩. Acesso em: 04 de out. de 2022.

21 NUMFOCUS, INC. *Pandas.get_dummies*. 2022. Documentação do método get_dummies da biblioteca Pandas. Disponível em: ⟨https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.get_dummies.html⟩. Acesso em: 12 de set. de 2022.

22 NUMFOCUS, INC. *Pandas.corr*. 2022. Documentação do método corr da biblioteca Pandas. Disponível em: ⟨https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.corr.html⟩. Acesso em: 12 de set. de 2022.

23 SCIKIT-LEARN DEVELOPERS. *sklearn.feature_selection.RFECV*. 2022. Documentação do método RFECV contido na biblioteca Scikit-Learn para Python. Disponível em: ⟨https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFECV.html⟩. Acesso em: 20 de nov. de 2022.

24 SCIKIT-LEARN DEVELOPERS. *Getting Started*. 2022. Documentação da biblioteca Scikit-Learn para Python. Disponível em: ⟨https://scikit-learn.org/stable/getting_started.html⟩. Acesso em: 20 de nov. de 2022.

25 SCIKIT-LEARN DEVELOPERS. *sklearn.ensemble.RandomForestClassifier*. 2022. Documentação do método RandomForestClassifier contido na biblioteca Scikit-Learn para Python. Disponível em: ⟨https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html⟩. Acesso em: 20 de nov. de 2022.

26 SCIKIT-LEARN DEVELOPERS. *sklearn.neighbors.KNeighborsClassifier*. 2022. Documentação do método KNeighborsClassifier contido na biblioteca Scikit-Learn para Python. Disponível em: ⟨https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html⟩. Acesso em: 20 de nov. de 2022.

27 SCIKIT-LEARN DEVELOPERS. *sklearn.model_selection.GridSearchCV*. 2022. Documentação do método GridSearchCV contido na biblioteca Scikit-Learn para Python. Disponível em: ⟨https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV⟩. Acesso em: 20 de nov. de 2022.

28 SCIKIT-LEARN DEVELOPERS. *sklearn.neighbors.DecisionTreeClassifier*. 2022. Documentação do modelo DecisionTreeClassifier contido na biblioteca Scikit-Learn para Python. Disponível em: ⟨https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html⟩. Acesso em: 20 de nov. de 2022.

29 THE IMBALANCED-LEARN DEVELOPERS. *Getting Started*. 2022. Documentação para instalação da biblioteca Imbalanced-Learn para Python. Disponível em: ⟨https://imbalanced-learn.org/stable/install.html#install⟩. Acesso em: 20 de nov. de 2022.

30 THE IMBALANCED-LEARN DEVELOPERS. *SMOTE*. 2022. Documentação para da função SMOTE da biblioteca Imbalanced-Learn para Python. Disponível em: ⟨https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html⟩. Acesso em: 20 de nov. de 2022.

31 CAMBRIDGE UNIVERSITY PRESS. *Queries as vectors*. 2008. Uso da distância cossenoidal em ferramentas de buscas. Disponível em: ⟨https://nlp.stanford.edu/IR-book/html/htmledition/queries-as-vectors-1.html⟩. Acesso em: 20 de nov. de 2022.

32 THE SCIPY COMMUNITY. *scipy.spatial.distance.cdist*. 2022. Documentação do método cdist da biblioteca Scipy em Python. Disponível em: ⟨https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.cdist.html⟩. Acesso em: 20 de nov. de 2022.

33 PANDAS. *Getting started*. 2022. Documentação para apresentação inicial da biblioteca Pandas para Python. Disponível em: ⟨https://pandas.pydata.org/getting_started.html⟩. Acesso em: 20 de nov. de 2022.

34 HRUSCHKA, E. R. *Aprendizado de Máquina - Classificação*. 2021. Breve explicação entre os diferentes conjuntos de treino, validação e teste. Disponível em: ⟨https://edisciplinas.usp.br/mod/resource/view.php?id=3737496⟩. Acesso em: 20 de nov. de 2022.

35 SCIKIT-LEARN DEVELOPERS. *sklearn.metrics.confusion_matrix*. 2022. Documentação do método confusion_matrix contido na biblioteca Scikit-Learn para

Python. Disponível em: ⟨https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html⟩. Acesso em: 20 de nov. de 2022.

36 SCIKIT-LEARN DEVELOPERS. *sklearn.metrics.classification_report*. 2022. Documentação do método classification_report contido na biblioteca Scikit-Learn para Python. Disponível em: ⟨https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html⟩. Acesso em: 20 de nov. de 2022.

37 SCIKIT-LEARN DEVELOPERS. *sklearn.preprocessing.Normalize*. 2022. Documentação do método Normalize da biblioteca SKLearn em Python. Disponível em: ⟨https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.normalize.html⟩. Acesso em: 04 de out. de 2022.

38 SCIKIT-LEARN DEVELOPERS. *Feature importances with a forest of trees*. 2022. Método *feature-importance* do modelo *Random Forest*. Disponível em: ⟨https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html⟩. Acesso em: 01 de nov. de 2022.

39 TSURUDA, A. L.; IVANO, C. M.; LOPEZ, V. C. *Landing Page Projeto*. 2022. Escola Politécnica da USP. Disponível em: ⟨https://sites.google.com/usp.br/tcc-analise-empresas-poli22/⟩. Acesso em: 30 de nov. de 2022.

40 TSURUDA, A. L.; IVANO, C. M.; LOPEZ, V. C. *Vídeo demonstração do projeto*. 2022. Escola Politécnica da USP. Disponível em: ⟨https://youtu.be/L-_wm1yAV84⟩. Acesso em: 07 de dez. de 2022.

41 NUMFOCUS, INC. *Pandas*. 2022. Biblioteca Pandas. Disponível em: ⟨https://pandas.pydata.org/⟩. Acesso em: 07 de set. de 2022.

42 SOUZA, S. N. A. de. *PCS3413 - Engenharia de Software e Banco de Dados (2019)*. 2019. Página web da disciplina PCS3413. Disponível em: ⟨https://edisciplinas.usp.br/course/view.php?id=67522⟩. Acesso em: 11 de dez. de 2022.

43 SNOWFLAKE, INC. *Streamlit Cloud*. 2022. Streamlit Cloud. Disponível em: ⟨https://streamlit.io/cloud⟩. Acesso em: 28 de nov. de 2022.

44 G40 TREINAMENTOS E CURSOS LTDA. *Product-Market Fit: definição, exemplos e como encontrar o seu*. 2022. Product Market Fit. Disponível em: ⟨https://g4educacao.com/portal/product-market-fit/⟩. Acesso em: 04 de dez. de 2022.

45 HUBSPOT, INC. *Quais são os benefícios do CRM para o marketing digital e as vendas online?* 2022. CRM. Disponível em: ⟨https://br.hubspot.com/blog/marketing/crm-marketing⟩. Acesso em: 04 de dez. de 2022.

46 ROCKCONTENT. *20 canais de comunicação diferentes para que você possa estar em contato com seus leads e clientes*. 2022. Canais de comunicação e marketing. Disponível em: ⟨https://rockcontent.com/br/blog/canais-de-comunicacao/⟩. Acesso em: 04 de dez. de 2022.

APÊNDICE A – LINK PARA REPOSITÓRIO GITHUB

O projeto completo pode ser acessado pelo link ⟨https://github.com/camilamiwa/tcc_publico⟩. Consulte o arquivo README.md para mais detalhes sobre o repositório.

APÊNDICE B – ENTREVISTA GUILHERME PASSOS E RENAN OLIVEIRA - ÂNIMA INVESTIMENTOS

Reunião com a Anima Investimentos - 08/04/2022

Presentes: Aline, Camila, Vinícius (alunos) e Victor (co-orientador); e Guilherme Passos e Renan Oliveira (Anima Investimentos)

Objetivo: coletar feedback deles enquanto investidores

- Apresentação do tema macro do trabalho proposto:
    - Os alunos explicaram as motivações do tema e as dores identificadas tanto do lado do investidor quanto dos empreendedores; deixando em aberto qual a preferência a fim de coletar feedbacks dos entrevistados sobre tais abordagens.
    - Também foi esclarecido o momento do trabalho, que à época estava em organização da ontologia e estruturação do banco de dados
- Feedback dos entrevistados:
    - Guilherme explicou que atualmente a Anima Investimentos não estava procurando novos aportes, mas sim focando em conduzir os negócios já investidos e como melhor desenvolver e alavancar tais negócios. Ainda assim, lembrou de um empecilho que sentia ao analisar projetos e pitch-decks: identificar aderência as suas premissas
    - Eles citaram que uma ferramenta que pudesse fazer uma triagem rápida e devolver apenas negócios que tivesse um mínimo de compatibilidade com suas teses: como área de aplicação e estágio de crescimento
- Próximos passos:
    - Que o grupo continue o desenvolvimento, e no próximo milestone, acione o Guilherme e o Renan para uma nova conversa e receber insights do direcionamento.

Duração: das 14h às 15h (1 hora)

APÊNDICE C – ENTREVISTA GUILHERME PASSOS E RENAN OLIVEIRA - ÂNIMA INVESTIMENTOS

Reunião com a Anima Investimentos - 13/12/2022

Presentes: Aline, Camila, Vinícius (alunos) e Victor (co-orientador); e Guilherme Passos e Renan Oliveira (Anima Investimentos)

Objetivo: coletar feedback como possíveis usuários do sistema desenvolvido

- Relembrar o tema macro do trabalho proposto:
    - Os alunos explicaram as motivações do tema, as dores identificadas tanto do lado do investidor quanto dos empreendedores e apresentaram a solução proposta com o projeto desenvolvido.
- Detalhar o projeto desenvolvido:
    - O grupo comentou sobre o processo de desenvolvimento do projeto;
    - Em sequência, apresentou o sistema atual, mesclando detalhes sobre a interface com o usuário e a lógica implementada em cada etapa.
- Feedback dos entrevistados:
    - Guilherme levantou pontos sobre a relevância do usuário poder personalizar suas teses, ou busca por investidores. Por exemplo, existem empreendedores que preferem receber aportes de fundos com mais experiência de mercado, e do lado do investidor, é de comum sabedoria que rentabilidade passada não é garantia de rentabilidade futura, por isso os investidores podem querer ponderar mais os dados mais recentes em sua base. O Guilherme enfatizou que, principalmente nesse mercado estudado, as tendências mudam com rapidez, por isso é interessante poder dar mais relevância aos registros mais atuais;
    - Renan comentou sobre a importância competitiva de uma boa base de dado, pensando principalmente em aplicações no mercado real;
    - Guilherme e Renan deram algumas sugestões para as próximas apresentações, como apresentar logo no começo o objetivo da reunião e entender o público alvo da reunião, neste caso, por exemplo, nesta situação não era necessário discorrer muito sobre os detalhes técnicos.

Duração: das 18h às 19h (1 hora)