

**Controle do Documento**

**Histórico de revisões**

| **Data** | **Autor** | **Versão** | **Resumo da atividade** |
| --- | --- | --- | --- |
| 09/08/2022 | João Alcaraz | 1.0 | Criação do documento |
| 09/08/2022 | Alexandre Fonseca, Gabriela Morais, João Alcaraz, Filipi Kikuchi, Lucas Pereira e Sofia Pimazzoni | 1.1 | Introdução, Análise SWOT e Value Proposition Canvas do produto |
| 09/08/2022 | Alexandre Fonseca, Gabriela Morais, João Alcaraz, Filipi Kikuchi, Lucas Pereira e Sofia Pimazzoni | 1.2 | Contexto da indústria e Matriz de Riscos |
| 11/08/2022 | Bruno Meira, João Alcaraz e Filipi Kikuchi | 2.0 | Descrição dos dados |
| 11/08/2022 | Gabriela Morais, Lucas Pereira e Sofia Pimazzoni | 2.1 | Primeira versão dos gráficos da relação entre as variáveis do gráfico. |
| 12/08/2022 | Bruno Meira, Gabriela Morais João Alcaraz, Filipi Kikuchi, Lucas Pereira e Sofia Pimazzoni | 2.2 | Objetivos e Justificativa, Descrição dos dados a serem utilizados, Descrição estatística básica dos dados e Descrição da predição desejada. |
| 26/08/2022 | Alexandre Fonseca, Bruno Meira, Gabriela Morais, João Alcaraz, Filipi Kikuchi, Lucas Pereira e Sofia Pimazzoni | 3.0 | Processo de categorização e formatação dos dados. |
| 05/09/2022 | Alexandre Fonseca, Bruno Meira, Gabriela Morais, João Alcaraz, Filipi Kikuchi, Lucas Pereira e Sofia Pimazzoni | 4.0 | Correções dos tópicos 1 ao 3 com relação ao feedback das sprints, modelagem dos dados e avaliação dos modelos de predição. |
| 09/09/2022 | Alexandre Fonseca, Bruno Meira, Gabriela Morais, João Alcaraz, Filipi Kikuchi, Lucas Pereira e Sofia Pimazzoni | 4.1 | Finalização das mudanças propostas na versão 4.0 |
| 23/09/2022 | Alexandre Fonseca, Bruno Meira, Gabriela Morais, João Alcaraz, Filipi Kikuchi, Lucas Pereira e Sofia Pimazzoni | 5.0 | Modelagem e avaliação dos algoritmos, hiperparâmetros dos algoritmos e parâmetros de treino e teste |

**Sumário**

[**1. Introdução**](#_2et92p0) **5**

[**2. Objetivos e Justificativa**](#_tyjcwt) **6**

[2.1. Objetivos](#_3dy6vkm) **6**

[2.2. Justificativa](#_1t3h5sf) 6

[**3. Metodologia**](#_4d34og8) **7**

[3.1. CRISP-DM](#_2s8eyo1) 7

[3.2. Ferramentas](#_17dp8vu) 7

[3.3. Principais técnicas empregadas](#_3rdcrjn) 7

[**4. Desenvolvimento e Resultados**](#_26in1rg) **8**

[4.1. Compreensão do Problema](#_lnxbz9) 8

[4.1.1. Contexto da indústria](#_35nkun2) 8

[4.1.2. Análise SWOT](#_1ksv4uv) 8

[4.1.3. Planejamento Geral da Solução](#_44sinio) 8

[4.1.4. Value Proposition Canvas](#_2jxsxqh) 8

[4.1.5. Matriz de Riscos](#_z337ya) 8

[4.1.6. Personas](#_3j2qqm3) 9

[4.1.7. Jornadas do Usuário](#_1y810tw) 9

[4.2. Compreensão dos Dados](#_4i7ojhp) 10

4.2.1 Descrição dos dados

[4.3. Preparação dos Dados](#_2xcytpi) 11

[4.4. Modelagem](#_1ci93xb) 12

4.4.1.Algoritmo com uso do Teorema de Bayes

4.4.2.Regressão logística

4.4.3.Algoritmo com uso do KNN

4.4.4.SVM

4.4.5.Random Forest

[4.5. Avaliação](#_2bn6wsx) 13

4.5.1.Métricas de avaliação

4.5.1.1.Acurácia

4.5.1.2.Recall

4.5.1.3.Precisão

4.5.1.4.Verdadeiros positivos e negativos

4.5.1.5.Falsos positivos e negativos

4.5.2.1.Avaliação do modelo com uso do Teorema de Bayes’ .

4.5.2.2.Avaliação da regressão logística.

4.5.2.3.Avaliação do modelo com KNN

4.5.2.4.Avaliação do SMV

4.5.2.5.Avaliação da random forest

[4.6 Comparação de Modelos](#_qsh70q) 14

[**5. Conclusões e Recomendações**](#_3as4poj) **14**

[**6. Referências**](#_1pxezwc) **15**

[**Anexos**](#_49x2ik5) **16**

**1. Introdução**

A Everymind é uma empresa de consultoria especializada na aplicação da Salesforce, utilizando diversas ferramentas e tecnologias para fornecer soluções personalizadas. Por ser uma das maiores na América Latina e integrar o grupo Uol, possui uma grande área de atuação, com presença nacional (15 estados brasileiros em 5 regiões) e internacional (escritórios no Japão e Europa). O problema expõe a preocupação da empresa com sua taxa de Turnover e na descoberta dos fatores que mais contribuem para a rotatividade de funcionários. Dessa forma, a empresa propõe a modelagem de um modelo preditivo que indique possíveis tendências de saída.

**2. Objetivos e Justificativa**

**2.1. Objetivos Geral**

A Everymind, como uma empresa que aplica o modelo Salesforce, possui o objetivo de gerenciar projetos dos seus clientes, a fim de aumentar a performance do negócio com excelência e qualidade. Através das reuniões e encontros com o parceiro, pudemos inferir alguns pontos relevantes. Devido à expansão da empresa no recente contexto pandêmico, ela conquistou seu espaço como um dos principais players no segmento. Dessa forma, manter a área de seus serviços expandida é de grande importância para o parceiro. Especificamente, a taxa de rotatividade da empresa é alta e entender os fatores que contribuem para a saída dos funcionários é um objeto de desejo interno da instituição.

**2.2 Objetivo Específico**

Proporcionar e expor informações mais detalhadas sobre os colaboradores que são mais propensos a sair ou que necessitem de uma ação de reconhecimento da Evermind com o objetivo de ter os colaboradores alinhados à cultura e estratégia da empresa, através da aplicação de variáveis formatadas e categorizadas do banco de dados dos colaboradores da empresa nos modelos preditivos de machine learning, sendo eles:

* K Nearest Neighbor
* Naive Bayes
* Árvore de decisão
* Support Vector Machine
* RandomForest
* Regressão Linear

No processo da mineração de dados e toda a estrutura do trabalho está sendo desenvolvida em um Notebook Oficial no Google Colaboratory, usando a metodologia CRISP-DM, com isso temos as etapas de todo o processo do trabalho bem definidas nos dando uma visão geral do ciclo de vida do processo de mineração de dados.

**2.3. Justificativa**

Fundada em 2014, a Everymind é uma das maiores parceiras Salesforce na América Latina com escritório no Brasil, além de atuações em implementações nas Américas, Japão e Europa. Atualmente com 280 colaboradores ativos, mais de 100 clientes ativos e com mais de 130 projetos em andamento, a Everymind enfrenta o problema de entender quais são os colaboradores mais propensos a saírem da empresa ou que necessitem de uma ação de reconhecimento.

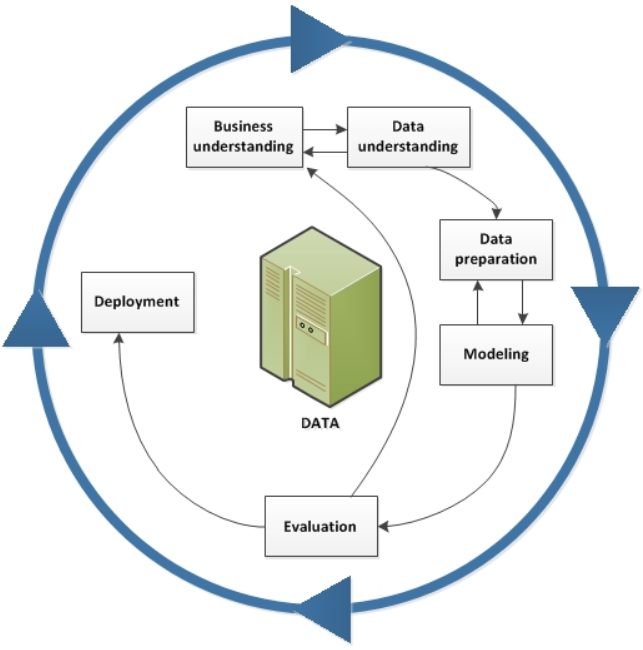
O presente trabalho irá apresentar uma forma de apontar quais são esses colaboradores com o maior fator de certeza para o apoio de decisão da Everymind, fazendo uso de modelos preditivos de machine learning criados especificamente para a base de dados dos colaboradores da empresa.

**3. Metodologia**

**3.1. CRISP-DM**

CRISP-DM é uma abreviação para Cross-Industry Standard Process for Data Mining (Processo Padrão Inter-Indústrias para Mineração de Dados, em tradução livre). Esta metodologia fornece uma visão geral do ciclo de vida do processo de mineração de dados, tendo fases bem definidas e complementares entre si.

O ciclo de vida do processo de mineração de dados possui 6 (seis) etapas bem definidas, sendo elas: 1. Entendimento do negócio; 2. Entendimento dos dados; 3. Preparação dos dados; 4. Modelagem; 5. Validação; 6. Implantação (do inglês, Deployment). Na Figura 1, as flechas indicam as dependências mais frequentes e importantes entre as fases.

**3.1.1. Fases do CRISP-DM**

1. Entendimento do negócio: Etapa na qual é necessário entender mais sobre o funcionamento das atividades do negócio e do problema a ser resolvido. Durante este processo, é fundamental conversar com pessoas diretamente envolvidas e estabelecer ideias concretas sobre os elementos que devem estar presentes na solução, incluindo as necessidades, expectativas, tecnologias e métodos estabelecidos.
2. Entendimento dos dados: Esta etapa acontece conjuntamente com a anterior e diz respeito à análise dos dados disponíveis e ao entendimento dos pontos fortes e das limitações que possuem. A partir destes, extrair informações e compreender sua confiabilidade e qualidade.
3. Preparação dos dados: Nesta etapa, acontece o pré-processamento dos dados, que os prepara e formata para que consigam ser utilizados por algoritmos. Durante esta fase, ocorre a seleção dos dados para análise e processos de limpeza, correção, adequação e derivação de novos atributos.
4. Modelagem: Aqui, serão escolhidas as técnicas mais adequadas para criar um modelo, baseando-se em testes iniciais de calibração dos parâmetros. Durante esta fase, pode ser necessário regressar à etapa anterior, visto que técnicas diferentes demandam formatos e conjuntos de dados distintos.
5. Avaliação: Processo no qual os resultados são validados, comparados com as expectativas criadas inicialmente e enfim aceitos ou enviados para reestruturação.
6. Implantação: Consiste no planejamento e implantação efetiva da solução, levando em consideração sua aderência às necessidades do negócio, seu nível de factibilidade, interpretabilidade e capacidade operacional.

**3.2. Ferramentas**

Para o desenvolvimento da solução, utilizamos o Google Colaboratory, ou simplesmente Colab, que é um serviço de nuvem gratuito para Aprendizado de Máquina e Inteligência Artificial. Dentro da interface da ferramenta, é possível adicionar código fonte (No caso, Python) e texto rico (em markdown) através de células, no formato Jupyter Notebook. As células podem ser executadas individualmente a fim conferir maior independência entre as partes e garantir testes unitários mais modularizados. A importação da base de dados é feita através do carregamento de um arquivo no próprio documento ou através de uma URL. Como o Colab roda em uma máquina do Google, não é necessário realizar configurações.

Para o gerenciamento de versões, utilizamos o Github, que é um serviço baseado em nuvem que hospeda um sistema de controle de versão (VCS) chamado Git. Dessa forma, é possível ter um histórico das modificações, o que facilita na manutenção e rastreamento de mudanças.

**3.3. Principais técnicas empregadas**

Para o tratamento e limpeza de dados, utilizamos diversas técnicas que auxiliaram na formatação e posterior utilização nos algoritmos. Entre elas, *label encoding* (transformação de classificações nominais por números)*,* formatação de espaços em branco (“ ”) entre palavras, padronização do formato de datas (YYYY-MM-DD) e transformação de strings em números inteiros.

Para a parte de modelos, utilizamos diversos algoritmos para auxiliar na predição de resultados a partir dos dados fornecidos. Entre eles podemos citar: KNN, Naive-Bayes, Árvore de Decisão, SVM, Random Forest e regressão logística. Todos estes algoritmos serão discutidos posteriormente na seção 4.4.

**4. Desenvolvimento e Resultados**

**4.1. Compreensão do Problema**

**4.1.1. Contexto da indústria**

*Pensando no melhor posicionamento e alinhamento da solução para alinhar a entrega com a expectativa da empresa, apresentaremos a seguir a a análise estratégica do cenário em que a solução irá atuar baseado nas 5 forças de porter, que são ameaça de produtos substitutos; ameaça de entrada de novos concorrentes; poder de negociação dos clientes; poder de negociação dos fornecedores e rivalidade entre os concorrentes.*

Players do mercado, de acordo com os sites “Baguete” e “imasters”: Dentre os principais players do mercado, estão as empresas: Imaginedone, SYS4B, JFox. São empresas de consultoria que fazem uso da Salesforce e estão, assim como a Everymind, em ascensão neste ramo da indústria.

O mercado de trabalho pós pandemia alterou o ambiente empresarial. O trabalho que antes era feito de forma presencial agora é feito de forma híbrida (em sua maioria) e homeoffice; com isso a comunicação entre funcionários é uma das principais preocupações visto que muitos dos colaboradores nem mesmo vivem no mesmo estado que a empresa. Além disso, o uso das nuvens para alocar dados é uma das principais tendências, afirma o site “Santodigital”.

Modelo de negócio: O objetivo é prever quais funcionários vão ficar e quais vão sair da empresa, analisando os dados coletados pelos stakeholders. Ao fazer a análise dos motivos causadores da alta rotatividade na empresa, o projeto espera melhorar o turnover de funcionários, além de atrair e manter mais funcionários na Everymind. Os recursos utilizados serão o Google Colab, o Python e o Pandas.

RIVALIDADE ENTRE OS CONCORRENTES

* Muitas empresas que prestam o mesmo serviço (Imaginedone, SYS4B,JFox,etc)

PODER DE BARGANHA ENTRE OS FORNECEDORES

* Migração dos fornecedores para ofertas mais vantajosas
* Poucos profissionais especializados em Salesforce

PODER DE BARGANHA DOS CLIENTES

* Clientes conseguem negociar e personalizar serviços com outros fornecedores
* Consulta fácil à outros serviços, permitindo comparações ágeis
* Cliente não é fiel a marca

AMEAÇA DE NOVOS ENTRANTES

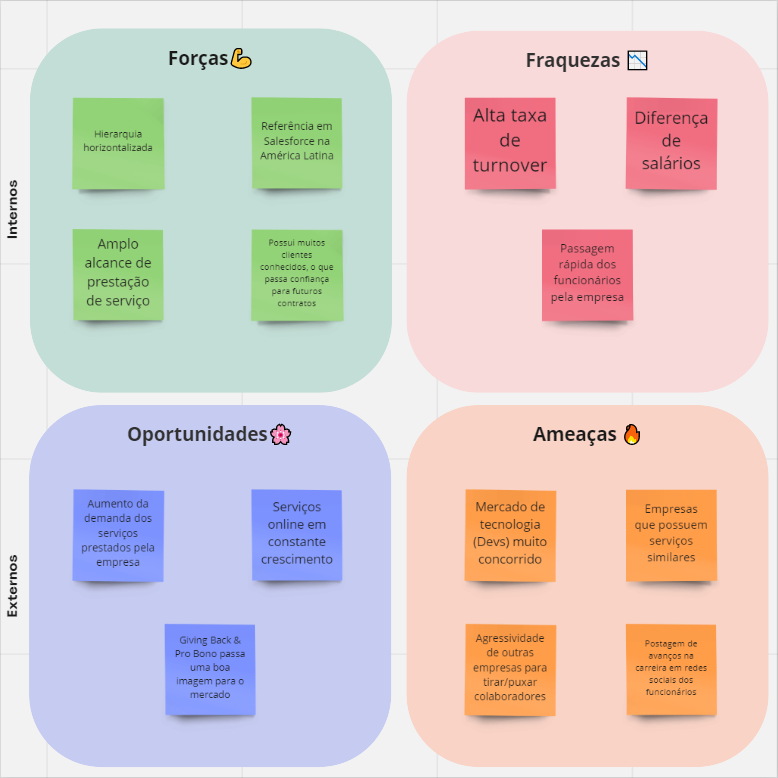
* Baixa barreira para empresas existentes entrar nesse mercado
* Alta demanda pelo serviço gera propostas melhores e diferentes

AMEAÇA DE NOVOS PRODUTOS OU SERVIÇOS SUBSTITUTOS

* Serviços com maior nível de personalização e eficiência
* Ofertas mais vantajosas

**4.1.2. Análise SWOT**

A análise SWOT é uma ferramenta que utiliza de quatro aspectos que ajudam a visualizar a posição de certa empresa no mercado. Os 4 aspectos são Forças, Fraquezas, Oportunidades e Ameaças, e são divididos entre, internos e externos, ou seja, se a empresa tem influência sobre tal fator do aspecto ou não.



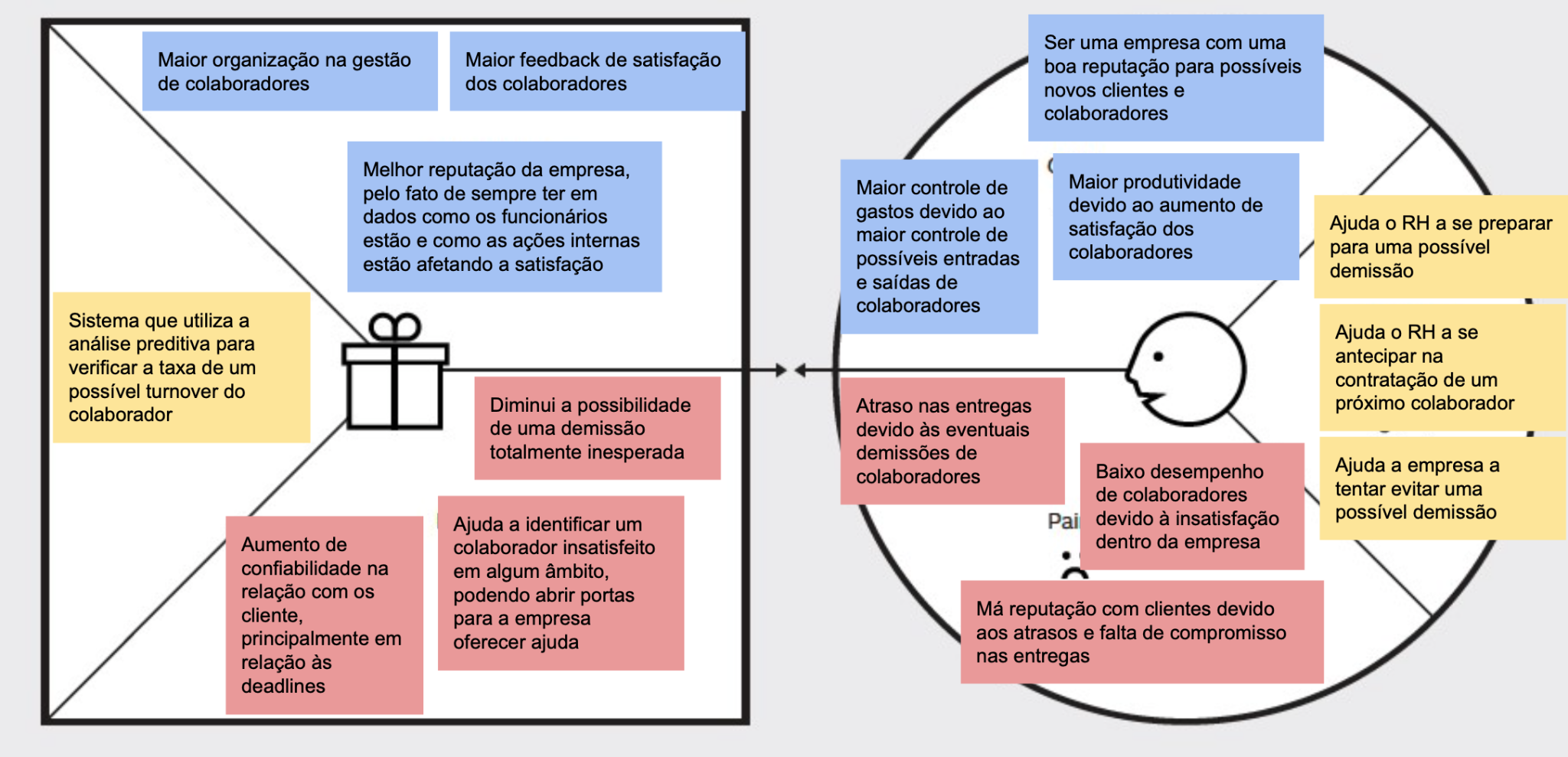
**4.1.3. Planejamento Geral da Solução**

A Everymind é uma empresa certificada SalesForce que está preocupada com sua alta taxa de Turnover e possui dificuldades em analisar os motivos responsáveis pela alta rotatividade. Neste contexto, a empresa forneceu dados referentes aos colaboradores (salário, cargo, data de entrada/saída, etc.) para que possamos desenvolver um modelo preditivo de classificação capaz de identificar quais funcionários têm tendências a sair ou permanecer no trabalho.

A solução proposta deverá ser usada como complementação do sistema atual pelo setor de RH, facilitando o processo, e como uma ferramenta de análise que possibilite a identificação de eventuais fatores que contribuem para uma demissão a fim de diagnosticá-los e contribuir para a tomada da melhor decisão possível. Consequentemente, isso será benéfico para a Everymind pois é melhor para uma empresa manter os funcionários a longo prazo. Dessa forma, serão usados dois critérios: a taxa de precisão do algoritmo e a taxa de Turnover da empresa para avaliar o desempenho da solução.

**4.1.4. Value Proposition Canvas**

O Value Proposition Canvas ajuda a empresa a pensar na organização das características da solução, expondo os seus diferenciais e priorizando as dores dos clientes. Por isso, criar uma Proposta de Valor é essencial, porque é a partir dela que o consumidor decide entre a nossa solução ou a de algum determinado similar.





**4.1.5. Matriz de Riscos**

A última ferramenta importante no entendimento de negócios é a matriz de risco. Sua função é ajudar a empresa a tomar decisões baseadas nos impactos e na probabilidade de certos riscos acontecerem, tanto com o projeto como com a empresa, sendo eles oportunidades ou não.

Abaixo é possível visualizar os riscos previstos e em seguida a matriz:

1- A empresa não fornecer os dados necessários

2- A AI não ser tão eficiente

3- Não conseguir fazer uma análise muito ampla dos dados

4- Vazamento dos dados

5- Não conseguir finalizar o protótipo

6- Melhor gestão de colaboradores

7- Diminuição de turnover

8- O grupo vai aprender como minerar e analisar dados

9- Baixo engajamento por parte dos envolvidos no projeto

10- Complexidade do projeto não mensurada

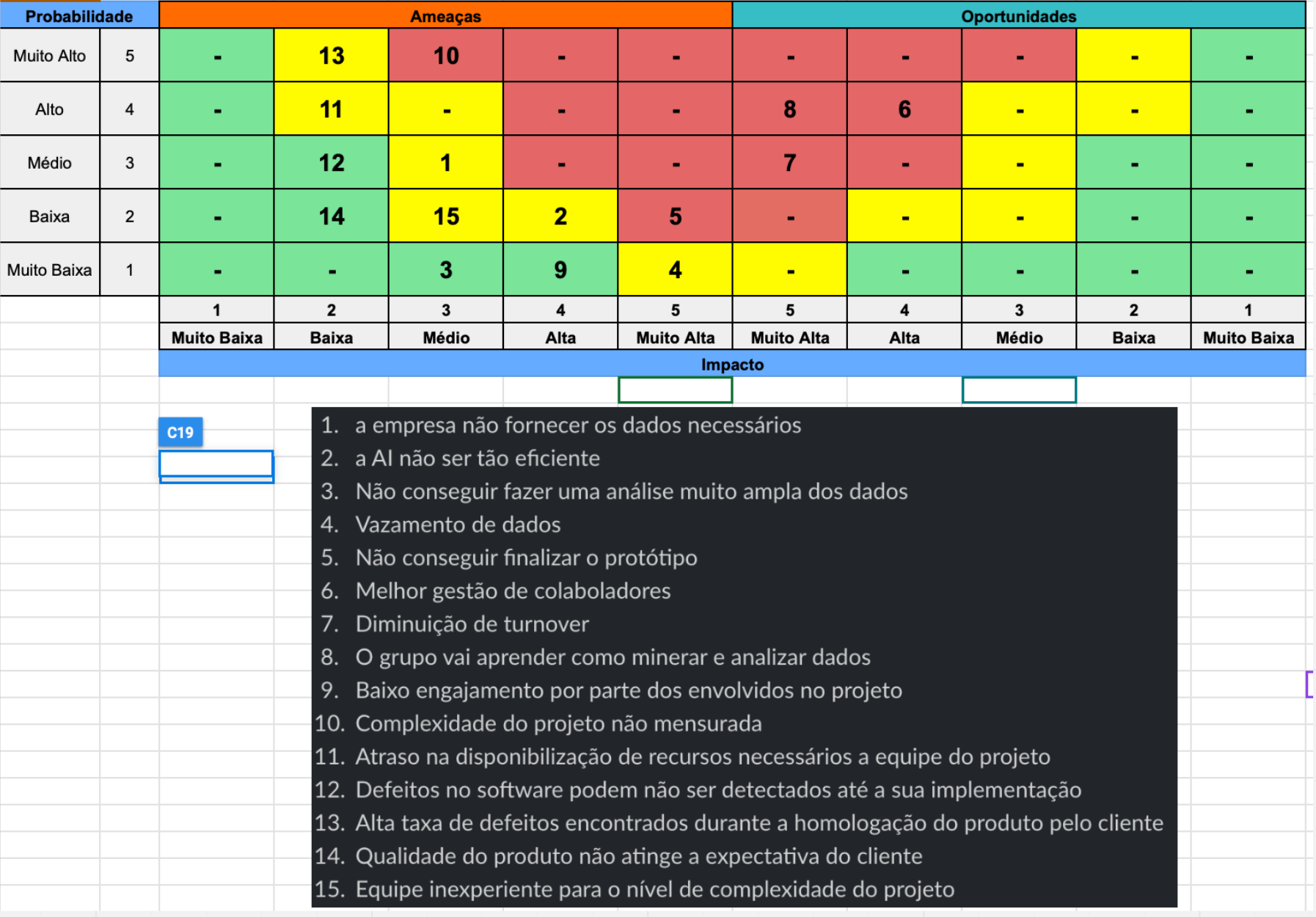
11- Atraso na disponibilização de recursos necessários a equipe do projeto

12- Defeitos no software podem não ser detectados até a sua implementação

13- Alta taxa de defeitos encontrados durante a homologação do produto pelo cliente

14- Qualidade do produto não atingir a expectativa do cliente

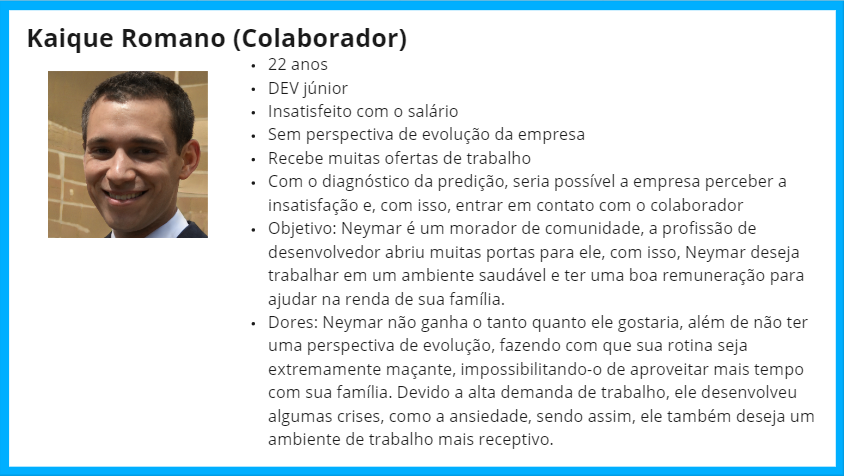
15- Equipe inexperiente para o nível de complexidade do projeto

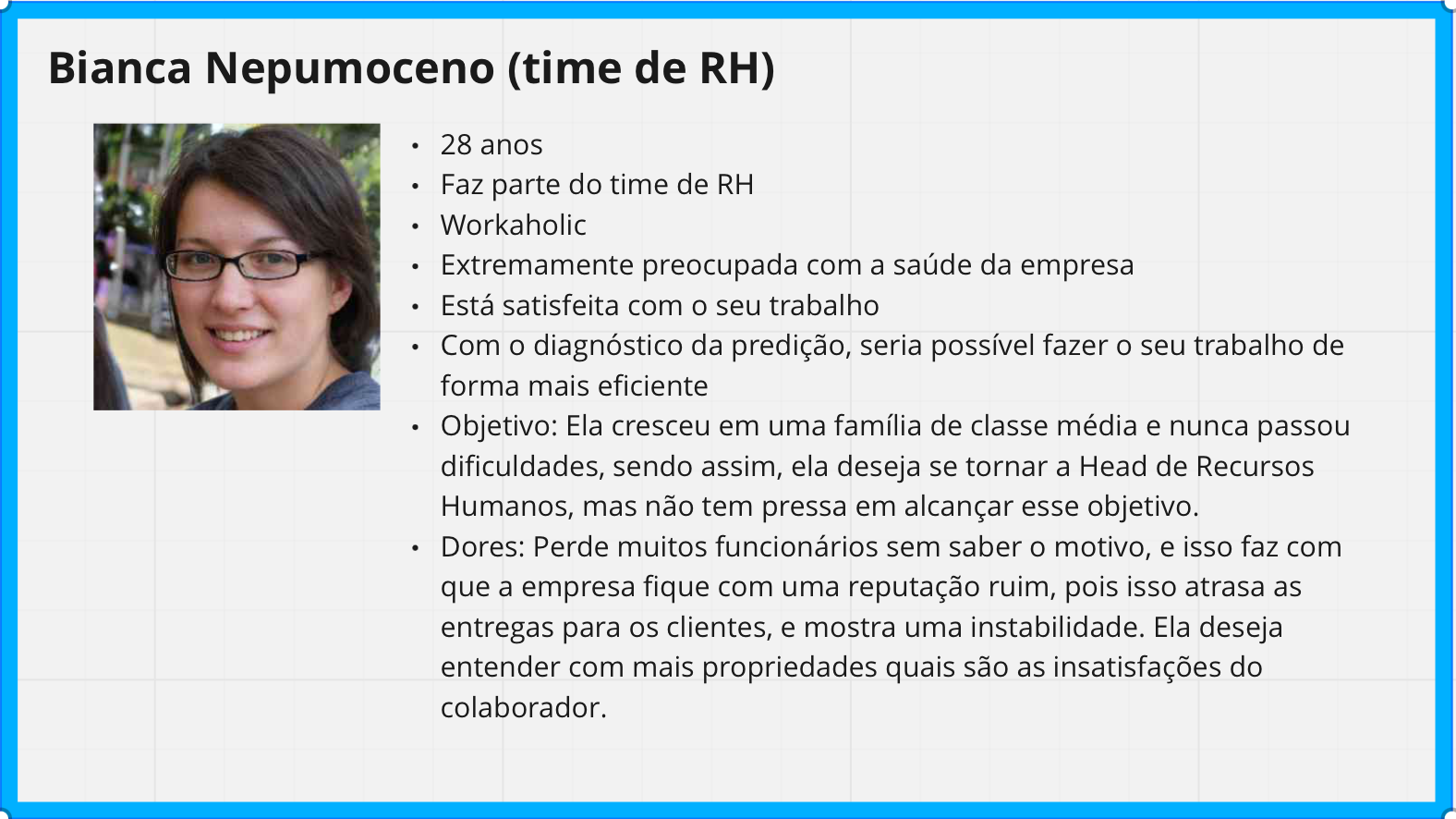


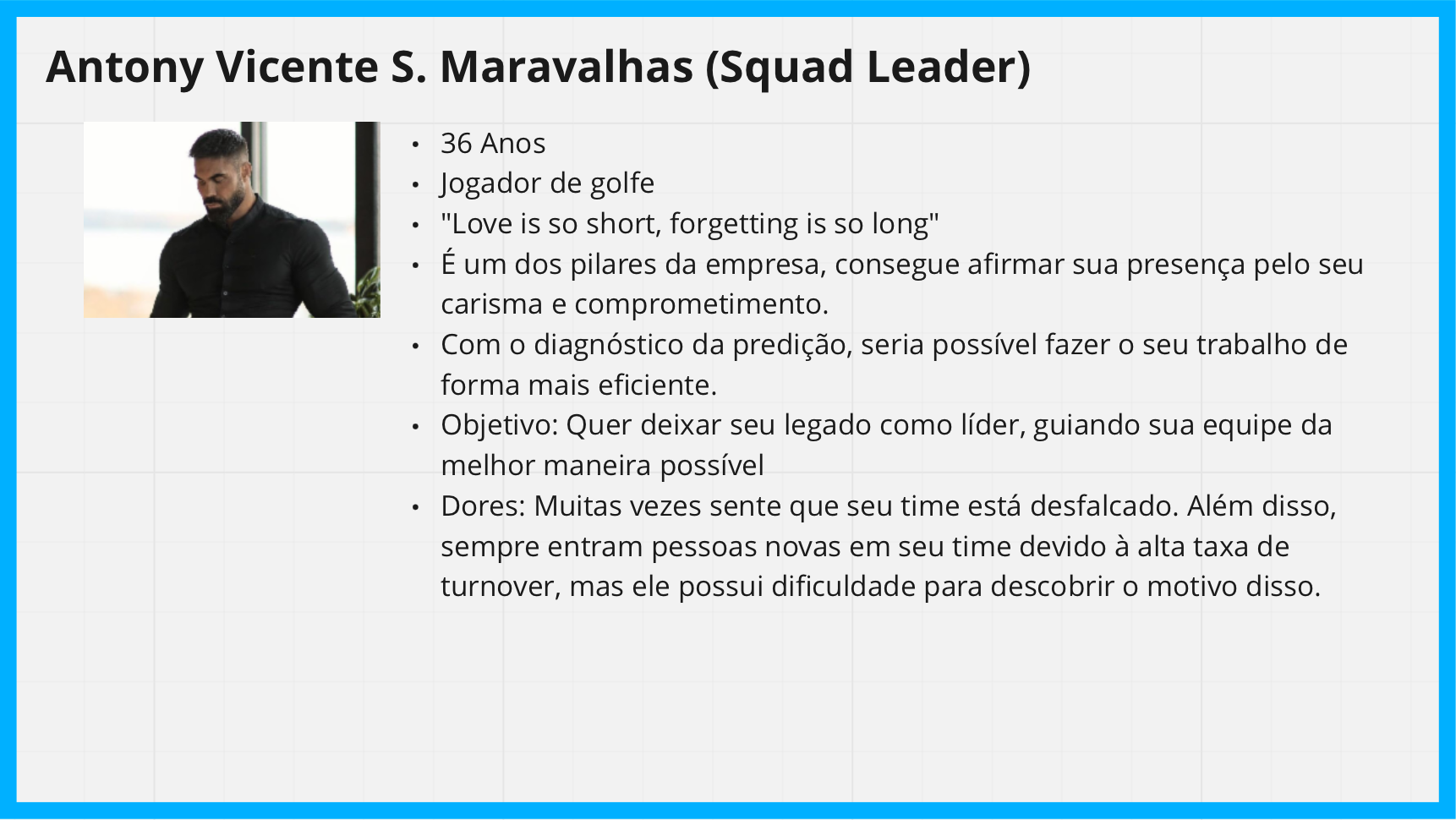
**4.1.6. Personas**

Persona é uma representação real do cliente do produto que vai ser desenvolvido. A persona tem um nome, idade, hobbies, um trabalho e mostra para a empresa para quem eles devem desenvolver o produto e onde devem focar para que ele ajude os clientes da melhor maneira possível.

Foram feitas três personas, uma para o time de administração da empresa, outra para os funcionários da mesma e a última para o TechLead da empresa. Abaixo é possível visualizar nossas personas:

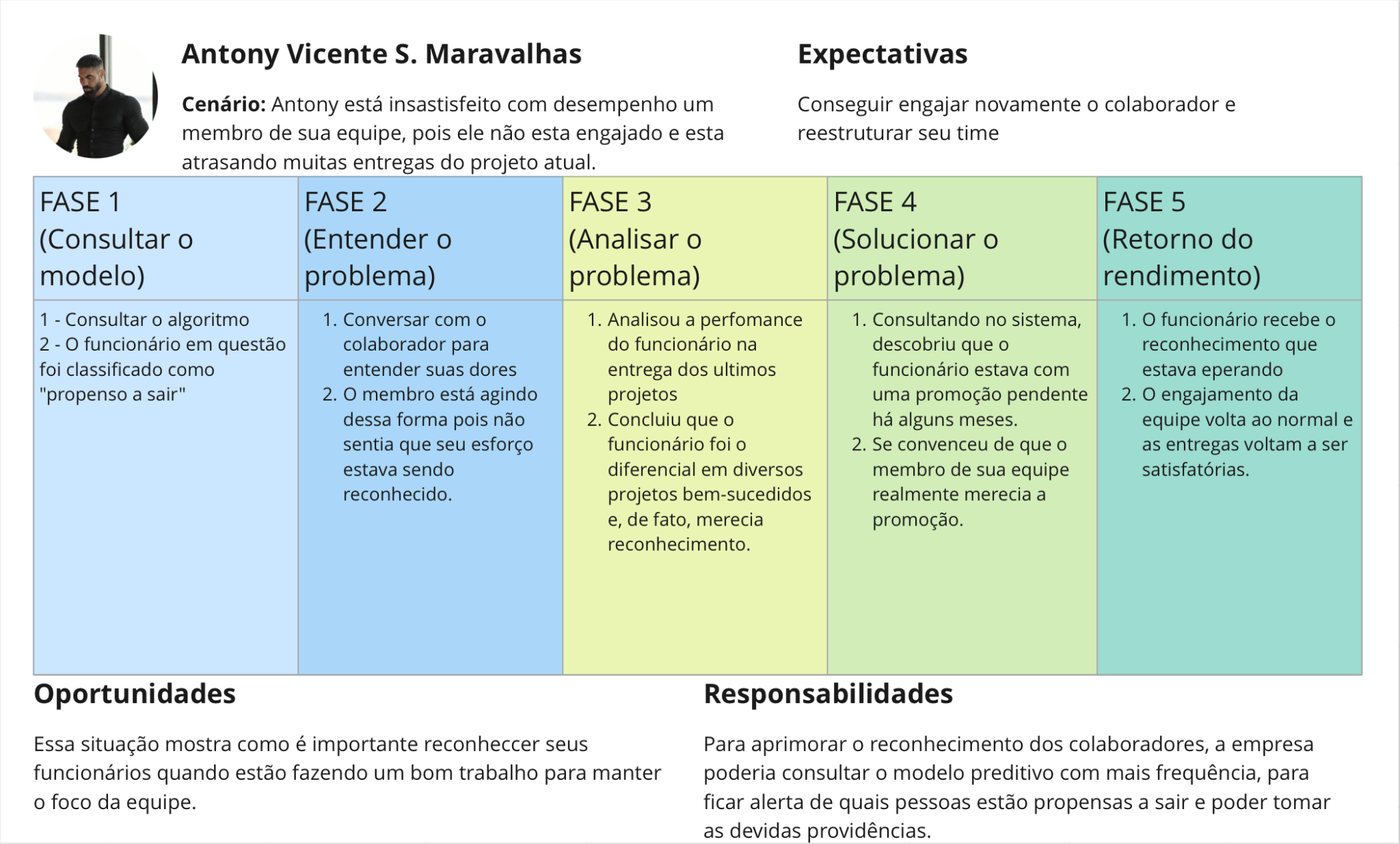
 

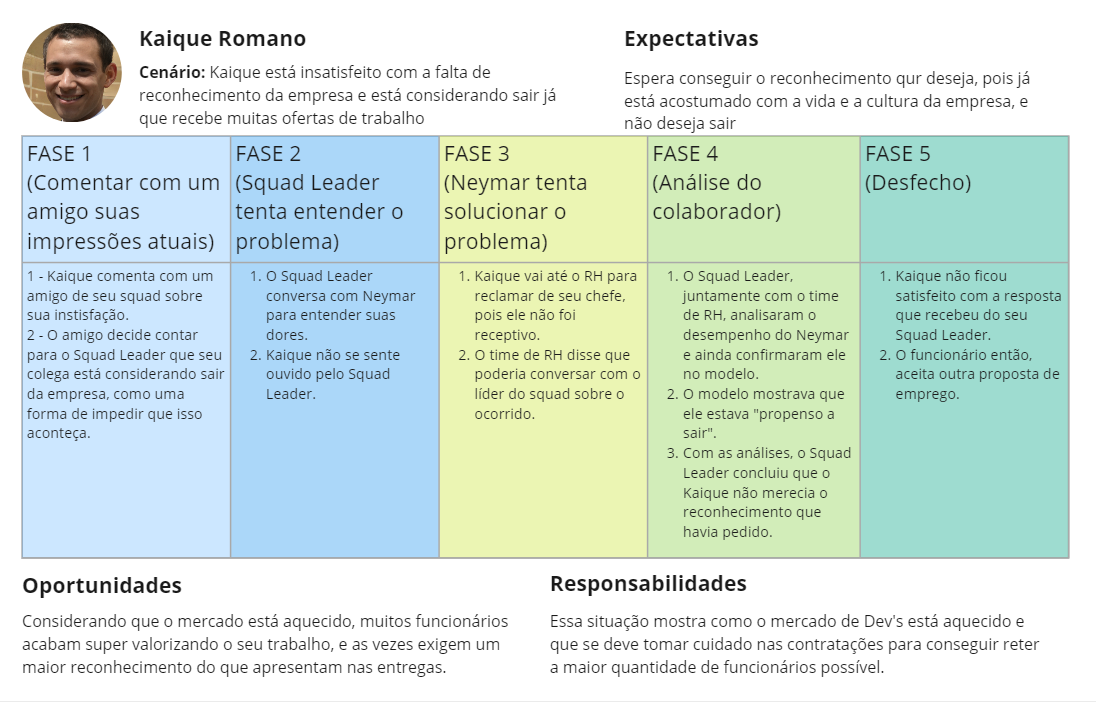




**4.1.7. Jornadas do Usuário**

A jornada de usuário consiste em um documento que apresenta, em ordem cronológica, o caminho que uma ou mais personas criadas para o projeto precisam percorrer para concluir determinada tarefa relacionada ao problema.





**4.2. Compreensão dos Dados**

**4.2.1 Descrição dos dados**

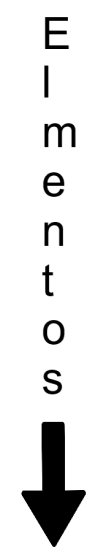
Nesta seção, destrinchamos os dados fornecidos do arquivo de dados “Base Colaboradores Everymind\_Inteli\_2020 a 2022vModelo Preditivo” fornecida na plataforma Google Sheets pelo parceiro, a partir do documento no formato .XLSX. Por se tratar de informações confidenciais, todos os dados da Everymind não podem ser divulgados e devem ser restritos ao compartilhamento aberto. Os dados abrangem todos os funcionários da empresa e descrevem cada um deles.

1. **Everymind**

Nesta seção, destrinchamos os dados fornecidos da sessão “Everymind” fornecida. Suas formatações e significados são apresentados abaixo.

Essa seção é a base de dados dos colaboradores em um arquivo .XLSX, fonte Calibri, contendo informações de 475 funcionários (475 linhas) divididas em 14 colunas. Para cada colaborador foi disponibilizado: situação(ativo, desligado ou afastado), data de admissão, data de saída e tipo de saída(caso o funcionário não esteja mais na empresa), cargo, salário mês(último salário pago), data de nascimento, gênero, etnia, estado civil, escolaridade(grau), área (área de atuação), cidade, estado e idade.

Os dados estão mascarados, logo não aparece nome, cpf e nenhum outro dado que identifique o colaborador, com algumas exceções como o VP, pois só tem um na empresa. Os salários, também não são reais, mas seguem a proporcionalidade real para que possa ser usado na solução.

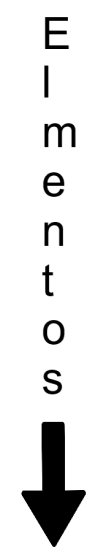
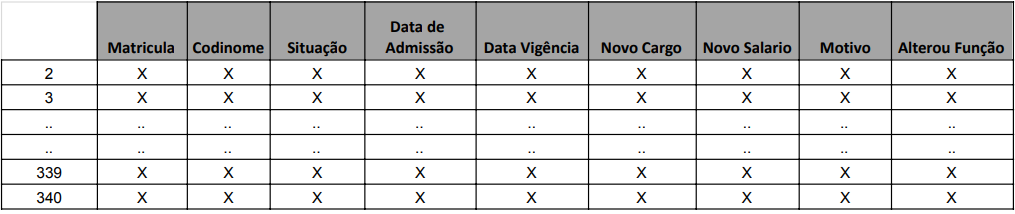




1. **Reconhecimento**

Nesta seção, destrinchamos os dados fornecidos da sessão “Reconhecimento” fornecida na plataforma Google Sheets pelo parceiro, a partir do documento no formato .XLSX. Suas formatações e significados são apresentados abaixo.







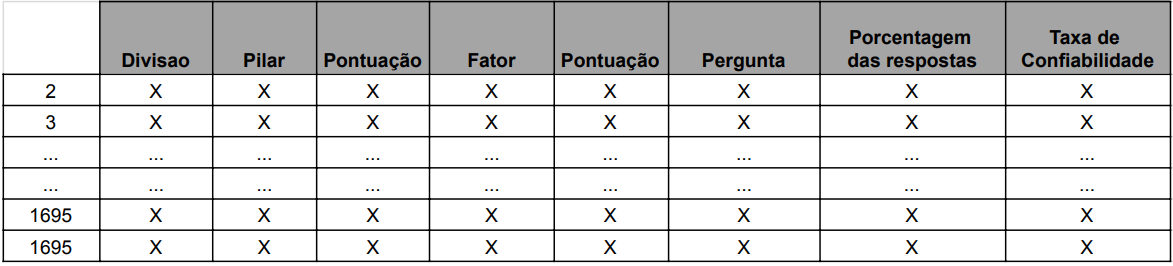
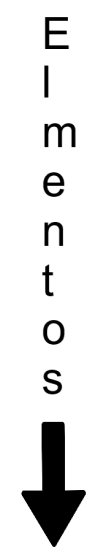
* Matrícula: A coluna contém o número de identificação do funcionário, definido por um número natural.
* Codinome: Esta coluna fornece o nome fictício do funcionário. O formato do dado é da forma “Pessoa Colaboradora ” seguida de um número natural.
* Situação: Descreve a situação atual do funcionário na empresa, podendo assumir estados: “Afastado”, “Ativo” ou “Desligado”.
* Data de Admissão: Informa a data de admissão do funcionário na empresa, no formato “MM/DD/AAAA”.
* Data Vigência: Define a data na qual o funcionário efetivamente começa a prestar seus serviços à empresa, apresentada no formato “DD/MM/AAAA”.
* Novo Cargo: Expressa o nome do novo cargo atribuído ao funcionário. Consiste em um texto que descreve a nova função. Na tabela fornecida, o valor da célula assume 26 valores. Exemplos: “Arquiteto Sr”, “Dev Jr”, "Líder IS”.
* Novo Salário: A coluna Novo Salário [sic] apresenta o valor do novo salário do funcionário. Assume o valor de um número com duas casas decimais.
* Motivo: Descreve a razão pela qual houve o remanejamento do cargo. O dado se apresenta na forma de um texto escrito em letras maiúsculas, tomando 3 (três) valores: “MÉRITO”, "PROMOÇÃO" e “RECLASSIF CARGO”.
* Alterou Função: Descreve se houve alteração de cargo, comparado ao anterior. O valor do dado é binário em texto, assumindo os valores “Sim” e “Não”.

1. **Ambiente de Trabalho**

Nesta seção, destrinchamos os dados fornecidos da sessão “Ambiente de Trabalho’’ fornecida na plataforma Google Sheets pelo parceiro, a partir do documento no formato .XLSX. Suas formatações e significados são apresentados abaixo.

A planilha de dados que foi disponibilizada é composta por colunas e linhas, cada linha contém os dados da pesquisa com um squad, ou seja, contendo um número de linhas iguais ao número de squad e as colunas contendo as variáveis estudadas.

Os dados dessa planilha são referentes ao estudo da Everymind de satisfação dos seus colaboradores no ambiente de trabalho, os dados são referentes a última pesquisa de satisfação realizada no dia 27/07/2022, e é feita entre todos os colaboradores de todos os setores a cada 3 meses.



* Divisão - Informa o setor do squad que respondeu a pesquisa, o valor da variável é do tipo string e assume onze (11) valores, exemplo: Mkt Cloud, People & Culture e Vendas.
* Pilar - Categoria da pesquisa, o valor da variável é do tipo string e assume dez (10) valores, exemplo: Relacionamento com o gestor, Vestir a camisa e Crescimento pessoal.
* Pontuação - Pontuação referente ao pilar, o valor da variável é do tipo number e é definido por um número natural de 1 (um) a 10 (dez).
* Fator - Subcategoria do pilar, o valor da variável é do tipo string e assume vinte e sete (27) valores, exemplo: Confiança no gestor, Orgulho e Propósito e Direcionamento.
* Pontuação - Pontuação referente ao fator, o valor da variável é do tipo number e é definido por um número natural de 1 (um) a 10 (dez).
* Pergunta - Pergunta feita ao colaborador, o valor da variável é do tipo string e assume trinta e três (33) perguntas diferentes.
* Porcentagem das respostas - Informa a porcentagem das respostas por squad, a variável é do tipo string.
* Taxa de confiabilidade - Valor referente a credibilidade da resposta do squad, a variável é do tipo string.

**4.2.1.1 Descrição do agrupamento e mescla**

Todos os dados serão analisados individualmente e, a partir do grau de importância de suas inter relações, utilizamos relações matemáticas pertinentes a fim de estabelecer informações que auxiliem na construção da solução.

**4.2.1.2 Descrição dos riscos e contingências**

Em relação aos riscos e pertinências, podemos estabelecer a qualidade dos dados por critérios estabelecidos pelo grupo a partir do grau de importância para a solução, sua profundidade/superficialidade e a cobertura/diversidade (quantidade de informações que podem ser inferidas deles). O acesso aos dados é limitado aos que foram disponibilizados na planilha e, em eventuais situações, podem ser adicionados conforme o acordo feito entre o grupo e o parceiro.

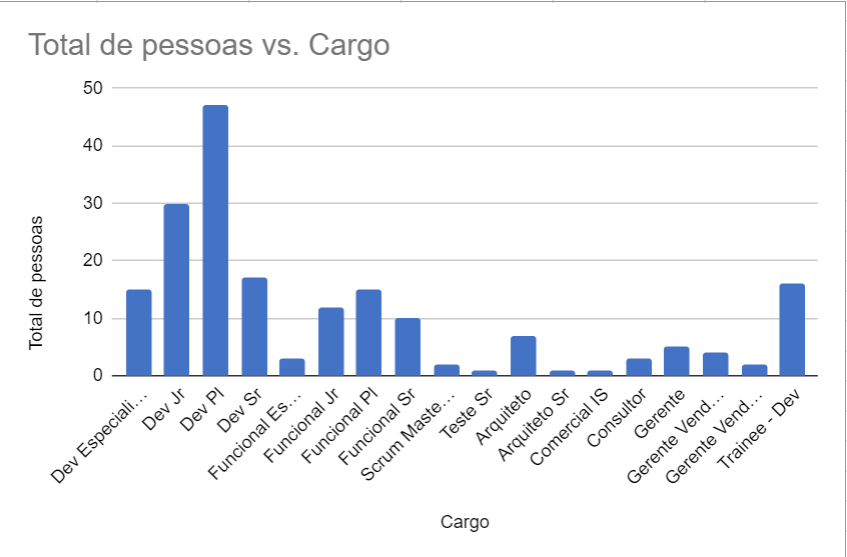
**4.2.1.3 Descrição dos criterios de escolha para análises iniciais**

As análises iniciais foram baseadas a partir da relação cargo x saída da empresa, a partir disso criamos algumas hipóteses, utilizando esses dados para relacionar com os méritos/promoções e o salário.(*Seção 4.2.2*)

**4.2.1.4 Descrição das restrições de segurança**

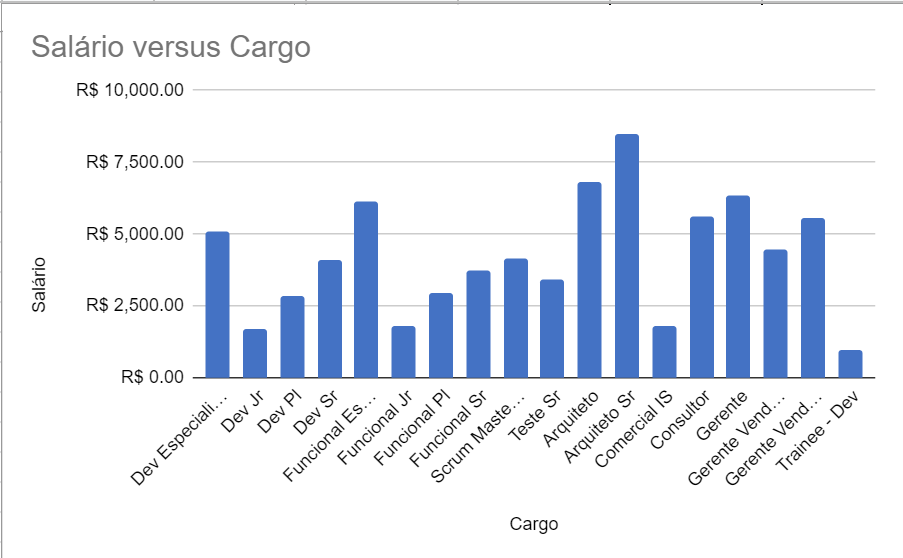
Por se tratar de informações confidenciais, todos os dados das bases de dados fornecidas não podem ser publicadas em nenhum lugar e os dados não podem ser divulgados.

**4.2.2 Descrição estatística básica dos dados**



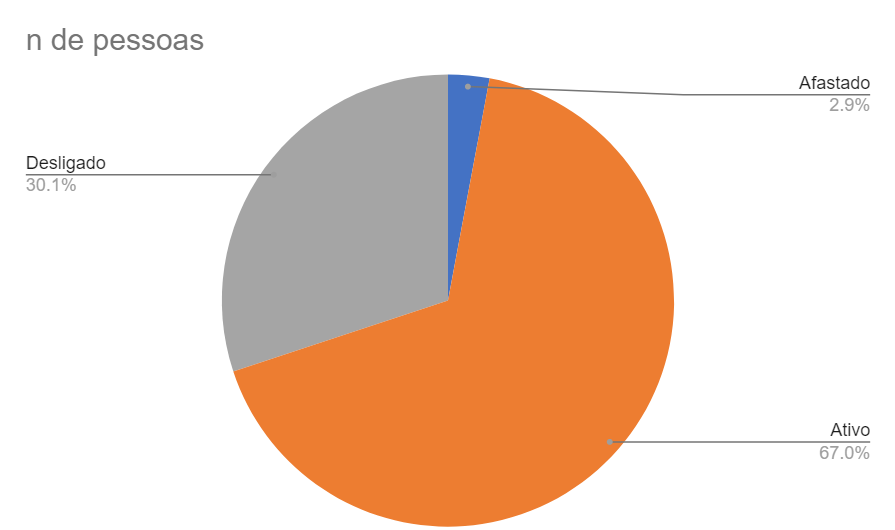
*gráfico 1 - total de pessoas que saíram pelo cargo que exerciam*

Como é possível visualizar no gráfico, os cargos com maior quantidade de pessoas são desenvolvedores, mas também são os que apresentam a maior taxa de demissão. A partir disso, criamos a hipótese de que o mercado de tecnologia é muito aquecido e esses funcionários provavelmente conseguiram oportunidades de trabalho que julgaram melhores.



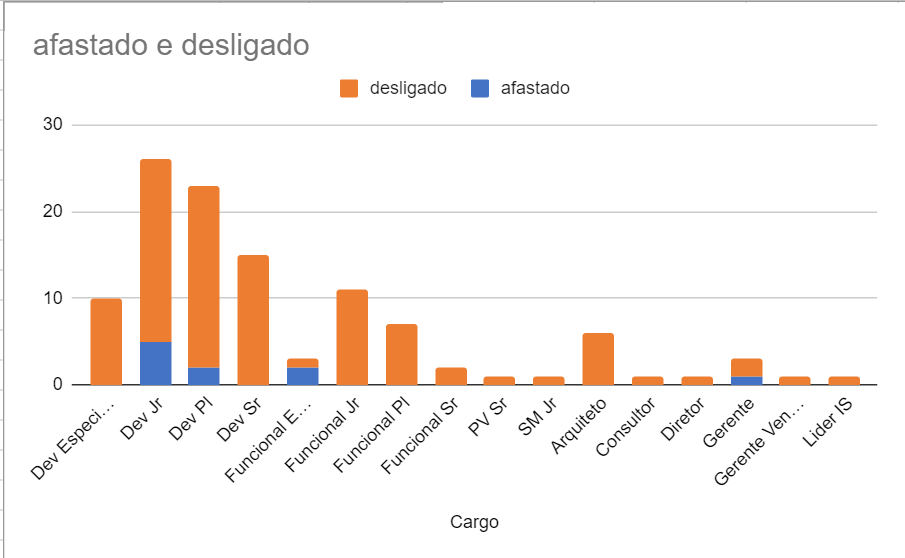
*gráfico 2 - média salarial das pessoas que saíram pelo cargo que exerciam*

Seguindo a hipótese do gráfico 1, pode-se fazer uma correlação com as informações apresentadas acima. Como é possível observar, a média do salário dos desenvolvedores era menor comparada a outros cargos, e isso pode ser outro fator que os fez sair da empresa, uma vez que eles poderiam ter recebido outras ofertas que consideraram ter uma melhor remuneração.



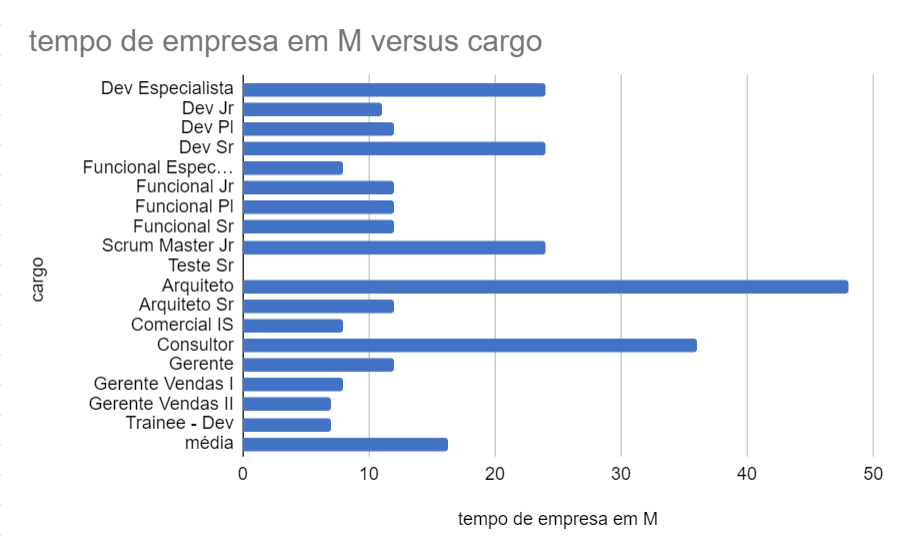
*gráfico 3 - comparação de pessoas que receberam promoções e permaneceram ou não na empresa*

A hipótese que pode ser extraída desse gráfico e que se relaciona com o gráfico 2, é que, considerando que ⅓ da empresa saiu mesmo após uma ou mais promoções, os funcionários continuam recebendo ofertas de trabalho que consideram ter uma melhor remuneração e, por isso, consideram sair da empresa mesmo após ter tido algum tipo de reconhecimento.



*gráfico 4 - comparação de pessoas que receberam promoções e saíram da empresa pelo cargo que exerciam*

O gráfico acima mostra que, das pessoas que receberam promoções e saíram (gráfico 3), a maioria são desenvolvedores, o que reforça a nossa hipótese de que os Devs costumam rotacionar no mercado com mais frequência.



*gráfico 5 - tempo médio de permanência de cada cargo na empresa*

No geral, juntando as análises de todos os gráficos, pode-se refletir que os desenvolvedores não permanecem muito tempo na empresa, uma vez que recebem muitas ofertas que julgam melhores. Por outro lado, os Devs não são os únicos que não permanecem muito tempo na empresa, e nossa hipótese para isso continua sendo o fato de os funcionários terem uma mente aberta para ofertas e oportunidades que julguem melhores.

**4.2.3 Descrição da predição desejada**

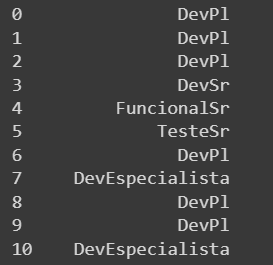
O modelo de predição será classificatório binário, ou seja, as classes já estão pré-definidas (“Saiu” ou “não saiu”) e o algoritmo vai definir qual colaborador se encaixa em cada classe. O algoritmo de aprendizado de máquina será supervisionado e diferentes algoritmos serão testados para que seja definido qual utilizar no modelo final.

**4.3. Preparação dos Dados**

A manipulação dos dados exige que eles estejam todos em formato de número (type: number) para fácil leitura e carregamento dos dados pelo algoritmo, os dados disponibilizados precisam ser passados por uma etapa de preparação. Essa etapa inclui tarefas de classificação e formatação de dados para modelagem, remover ou substituir registros em branco, seleção de um subconjunto de amostras para análise, derivação de novos atributos e mesclar conjuntos de dados e registros.

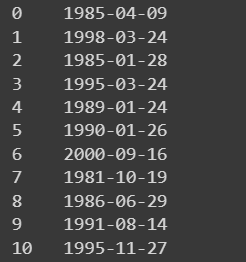
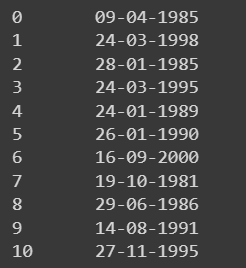
**4.3.1 Classificação e formatação de dados para modelagem**

Para a formatação inicialmente precisaremos tirar os espaços de toda a tabela a fim de padronizar todos os dados de todas as colunas, para isso fizemos a substituição de espaço (“ ”) para (“”), exemplo: “Superior incompleto” para “Superiorincompleto”. Essa Feature foi selecionada para possibilitar a utilização do Label Encoder e do One Hot Encoder para a categorização das informações. Um exemplo das transformações segue abaixo:

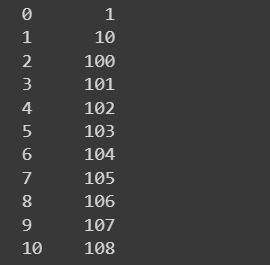
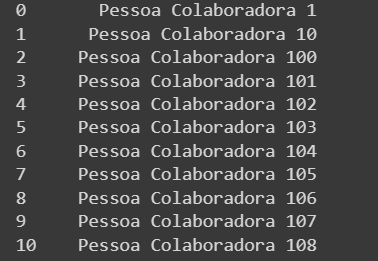
Durante esse processo vamos tratar os dados a fim de padronizá-los para que sejam aceitos e melhor utilizados pelo algoritmo a partir de funções que modificam a forma do dado. No momento, estamos trabalhando com alguns tipos de dados, sendo eles dados relacionados a tempo e dados relacionados a nome.

Nos dados relacionados à data foram formatados apenas a ordem de dd/mm/yyyy (Exemplo: 31/10/2003) para yyyy/mm/dd (Exemplo: 2003/10/31). Sofrem essa alteração os dados presentes na aba “Everymind” nas colunas "Dt Admissao", "Dt Nascimento" e "Dt Saida" e na aba “Reconhecimento” nas colunas “Data de Admissão” e “Data Vigência”. Essa Feature foi selecionada para padronizar todas as datas do banco e facilitar o cálculo entre duas datas para análises futuras. Um exemplo das transformações segue abaixo:

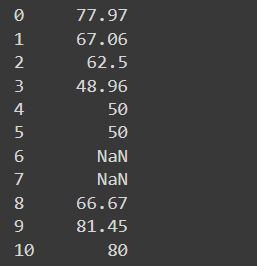
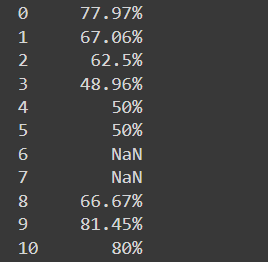


Nos dados relacionados a nome foram formatados os textos com o objetivo de permanecer apenas os números. Exemplo: “PessoaColaboradora197” foi formatado para apenas “197”. Sofrem essa alteração os dados presentes na aba “Everymind” nas coluna “Nome Completo” e na aba “Reconhecimento” na coluna “Codinome”. Essa Feature foi selecionada para padronizar todos os dados categóricos em números e facilitar análises futuras do algoritmo.

Um exemplo das transformações segue abaixo:



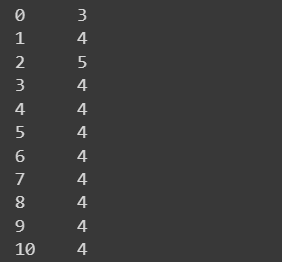
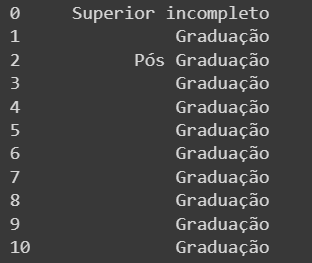
As colunas “Pulou”, “Muito Insatisfeito”, “Insatisfeito”, “Neutro”, “Satisfeito” e “Muito Satisfeito” da aba “Ambiente de Trabalho 27.07” mesmo estando em porcentagem o algoritmo reconhece como formato de texto (String) e para transformar em número (Number) trocando os (“%”) por (“”). Exemplo: “45,67%” foi transformado apenas para “45,67”. Essa Feature foi selecionada para padronizar todos os dados do banco no tipo número e facilitando o manuseio para análises futuras. Um exemplo das transformações segue abaixo:



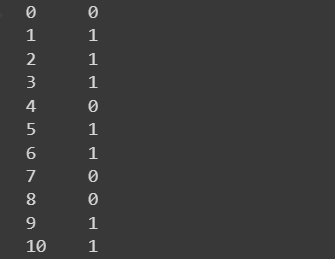
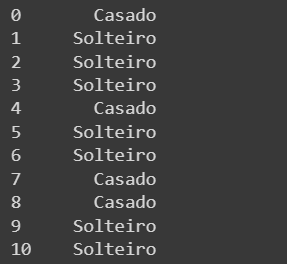
Grande parte dos dados são categóricos e quando temos categorias como descrição do dado

precisamos converter para valores numéricos, podemos fazer isso usando o Label Encoder que faz uma atribuição numérica crescente para cada categoria impondo uma ordenação entre as classes e o One Hot Encoder que cria uma coluna para cada valor e faz uma atribuição do valor 1(um) para a coluna correspondente da amostra e consequentemente não necessitando de uma ordenação.

As transformações usando o Label Encoder foi usado na tabela “Everymind” na coluna “Escolaridade ”, fizemos uma atribuição numérica dos dados da coluna em ordem crescente e ordinal. Exemplo: “EnsinoMédioIncompleto” atribui “0”, “EnsinoMédio” atribui “1”, etc. Essa Feature foi selecionada para transformar os dados categóricos em numéricos possibilitando a utilização deles em análises futuras . Um exemplo das transformações segue abaixo:



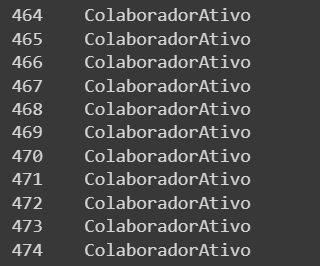
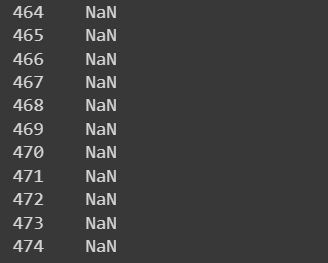
As transformações usando o One Hot Encoder foram usadas em todas as tabelas do banco de dados e em dezessete colunas no total, foi feito uma atribuição dos valores em colunas e uma atribuição de números (0 e 1) à essa colunas para indicar se a coluna é correspondente a amostra, exemplo: Os valores da coluna “Estado Civil” se transformaram em colunas e foi atribuído o número 1 (um) para correspondente e 0 (zero) para não correspondente, podendo só ter apenas um número 1(um) na linha. Essa Feature foi selecionada para transformar os dados categóricos em numéricos possibilitando a utilização deles em análises futuras. Um exemplo das transformações segue abaixo:



**4.3.2 Remover ou substituir registros em branco**

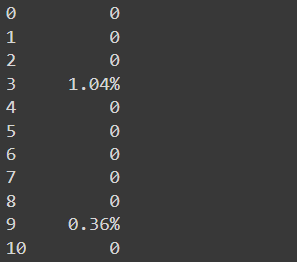
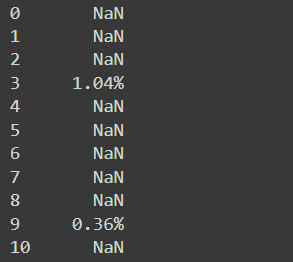
Em nosso modelo preditivo ter registros em brancos prejudica a análise do algoritmo, tendo isso em vista, detectamos que nas colunas “Dt Saida” e “Tipo Saida” da aba “Everymind” e as colunas “Pulou”, “Muito Insatisfeito", “Insatisfeito”, “Neutro”, “Satisfeito” e “Muito Satisfeito” da aba “Ambiente de Trabalho 27.07” haviam dados vazios e precisariam ser preenchidos. Essa Feature foi selecionada para os campos vazios na tabela não ocasionarem erros em nossa predição do algoritmo e prejudicar a confiabilidade das informações.

Da aba “Everymind”, a coluna “Dt Saida” estavam em formato de data e os valores vazios representavam que o colaborador daquela linha em específico ainda estava ativo na empresa, então apenas substituímos o valor de nulo para a data atual que se atualiza conforme os dias passam. A coluna “Tipo Saida” estava em formato de texto(string) e os valores vazios na coluna representam que o colaborador ainda está ativo na empresa, então substituímos o valor nulo para “ColaboradorAtivo”. Um exemplo das transformações segue abaixo:



Da aba “Ambiente de Trabalho 27.07”, como se tratavam de dados numéricos e os valores vazios representavam que aquela opção não foi escolhida por nenhum colaborador do setor que participou da pesquisa, então apenas substituímos o valor de vazio para o número 0 (zero).

Um exemplo das transformações segue abaixo:



**4.3.3 Seleção de um subconjunto de amostras para análise**

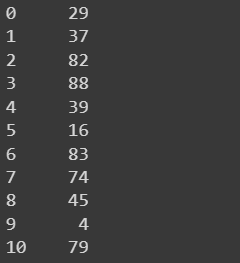
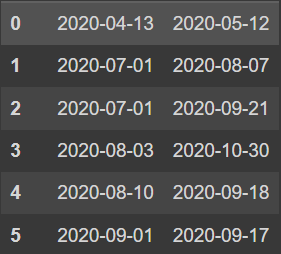
Dentro da tabela, fizemos uma seleção de amostra de todos os funcionários que saíram da empresa e criamos uma nova tabela apenas com esses dados. Essa Feature foi selecionada para dar um foco nos colaboradores inativos, facilitando a análise e decisão de quais fatores mais influenciam a decisão de deixar a empresa.

Funcionários que saíram em menos de um ano. Essa Feature foi selecionada para dar um foco nos colaboradores que saíram da empresa em menos de um ano dentre os inativos para investigar as variáveis que mais influenciam na decisão.

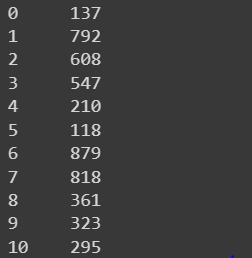
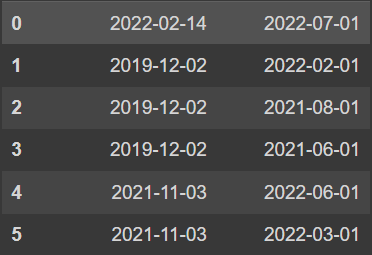
**4.3.4 Derivação de novos atributos**

Durante toda a formatação dos dados, foi detectada a necessidade de cálculos entre as datas da tabela. Essa Feature foi selecionada para facilitar a análise dos dados, deixando de lado a necessidade de fazer cálculos complexos com frequência.

Na aba “Everymind” fizemos o cálculo entre a data de admissão (“Dt Admissao”) e a data de saída (“Dt Saida”) para obter os meses de empresa e entre a data de nascimento (“Dt Nascimento”) e a data de saída (“Dt Saida”) para obter a idade dos colaboradores. Um exemplo dos cálculos segue abaixo:



Na aba “Reconhecimento” fizemos o cálculo entre a data de admissão (“Data de Admissão”) e a data de vigência (“Data Vigência”) para obter o número de dias entre a data de admissão e a data que o colaborador foi reconhecido na empresa. Um exemplo dos cálculos segue abaixo:



Também transformamos algumas colunas em novas colunas menores com dados que mais dão ganho de informação pro algoritmo como a coluna de cargos que agora só informa se o colaborador é ou não dev, foi feito também uma coluna de média salarial que diz se o colaborador está ou não acima da média salarial da empresa, e por último uma coluna se diz ou não se o colaborador saiu ou não da empresa. Todas essas novas informações registradas atribuem 1 para correspondente e 0 para não-correspondente. Essa feature foi selecionada para diminuir dados com intervalo de números muitos grande (Exemplo: Salário mês é um intervalo de 500 a 13.000) e diminuir números de colunas (Muitas colunas com cargos que não tem um ganho de informação relevante para o algoritmo), fazendo esses processos a avaliação do modelo tem uma grande avanço.

**4.3.5 Colunas não utilizadas**

Nesta etapa foi feita uma seleção dos dados e definição da relevância do atributo no nosso modelo, um atributo observado foi a questão da etnia, usar isso no modelo como fator decisivo é antiético, cria um viés negativo, deixa o modelo questionável com nível baixo de credibilidade e tira a viabilidade da solução, pensado nisso foi retirado da análise a coluna contendo a etnia dos colaboradores. Essa feature foi selecionada para analisar as variáveis que não tem validade para o resultado e restringi-la especificamente, já que não fazem sentido para o negócio.

**4.3.6 Mesclar conjuntos de dados e registros**

Ao fim da formatação, categorização e padronização do banco de dados, todas as informações foram transferidos para uma nova tabela em que o algoritmo poderá trabalhar com ela no backend. Foi criada uma nova tabela correspondente para cada aba da base de dados. Essa Feature foi selecionada para que o algoritmo possa fazer as análises em uma tabela com os dados formatados sem que isso altere a tabela original da empresa.

**4.3.7 Oversampling**

Oversampling é uma técnica usada para igualar a quantidade de dados de certas features. Por exemplo, caso for selecionada uma coluna para a análise, mas as informações presentes nela tem uma diferença de quantidade muito grande, é feito um processo para aumentar as informações da menor quantia, para que elas fiquem na iguais a de maior quantia. Dessa forma, o modelo poderá fazer a análise com mais dados e ter um resultado, na maioria das vezes, mais preciso.

No projeto foi utilizado esse processo em todas as colunas que apresentavam diferença na quantia dos dados, como por exemplo, a coluna "Dt Saida", uma vez que a quantidade de colaboradores que saiu da empresa é menor do que a que não saiu, e, ao aplicar o oversampling, essa quantia foi igualada.

A maioria dos modelos apresentou melhora após o uso dessa ferramenta, como por exemplo, o Random Forest, que sofreu um aumento de 18% e atingiu os 80%, o Naive Bayes, que estava com 63% e aumentou para 70%, o SVM manteve-se com uma acurácia de 65%. Porém, tiveram modelos que não apresentaram melhoria alguma, sendo eles: a Regressão logística, que estava com 74% e foi para 66%, o Knn que diminuiu de 69% para 63% e Árvore de decisão que diminuiu 2%.

**4.3.8 Normalização**

Normalizar os dados é deixá-los em uma mesma escala para que não haja problemas no treinamento do modelo. Existem vários tipos de normalização, mas a que foi usada em todos os modelos do projeto foi a MinMaxScaler(), que transforma os dados para que fiquem em um intervalo entre 0 e 1, assim fazendo com que não haja valores extremos.

**4.3.9 Padronização**

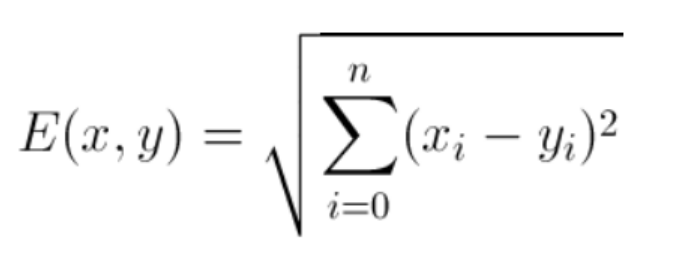
Para a aplicação de alguns modelos, é necessário fazer a padronização dos dados, que é semelhante à normalização e visa colocar os dados em uma escala, porém tornamos a médias dos dados zero (0) e o desvio padrão como (1). Essa transformação auxilia na performance de alguns modelos, como no SVM.

**4.4. Modelagem**

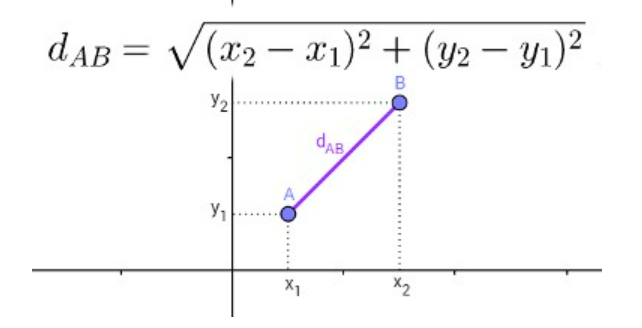
**4.4.1 K Nearest Neighbor**

O KNN é um algoritmo não paramétrico, onde a estrutura do modelo será determinada pelo dataset utilizado, ou seja, o conjunto de dados será determinado de acordo com a distância dos “vizinhos” mais próximos. Este algoritmo também é conhecido como de aprendizado lento ou melhor dizendo, é um algoritmo com um tempo de resposta mais demorado quando comparado aos demais, que também utilizam aprendizado supervisionado. Além disso, utiliza várias métricas para o cálculo da distância entre N dimensões, como o caso das distâncias de Hamming, Mahalanobis e Minkowski, porém a que utilizamos é a distância Euclidiana, que funciona da seguinte forma:

A métrica de distância simples trata-se da raiz quadrada da soma das diferenças de cada característica ao quadrado. É basicamente a distância simples entre pontos, utilizando algumas versões distintas, sendo cada versão uma base para a aplicação no treino. Seguem as fórmulas das aplicações:



**Fórmula geral da distância Euclidiana**

****

**Exemplificação de uma aplicação da distância Euclidiana**

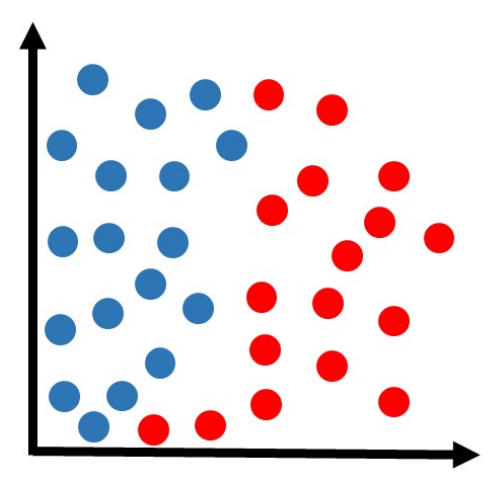
Exemplificando como acontece todo o processo de classificação utilizando o modelo KNN:

**1**) O modelo recebe um dado não classificado e mede a distância do novo dado em relação a cada um dos outros dados que já estão classificados;

**2** ) O modelo seleciona as K menores distâncias em relação ao novo dado;

**3**) Verifica a classe dos dados que tiveram as K menores distâncias e contabiliza a quantidade de vezes que cada classe que apareceu;

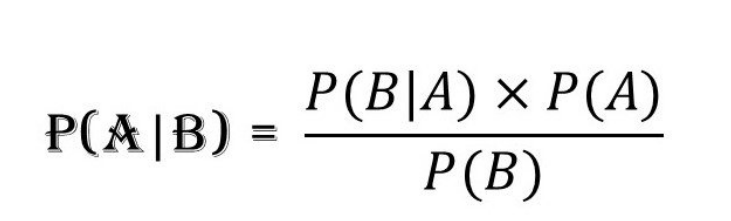
**4** )  Classifica esse novo dado como pertencente à classe que mais apareceu na análise completa.



Essa imagem é geralmente utilizada para representar o KNN de uma forma simples, inserindo um novo “dado” com classe desconhecida neste gráfico, por exemplo uma bola verde, o modelo KNN classificaria a bola com a cor vermelha ou azul, de acordo com a proximidade entre as bolas com classe já conhecida.

**4.4.2 Naive Bayes**

O algoritmo Naive Bayes é um classificador probabilístico que usa o Teorema de Bayes para categorizar textos baseado na frequência das palavras usadas. O modelo desconsidera as correlações entre as features, cria uma tabela de probabilidade (fórmula abaixo) e calcula a partir dela o que tem maior chance de ocorrer.



Para a criação do modelo usamos como atributos previsores no primeiro teste os seguintes dados, “Cargo”, “Estado civil”, “Escolaridade” e “Área”; e como variável alvo usamos a coluna nomeada como “Tipo saída” (caso seja true, a pessoa saiu ou tem tendência a sair, caso seja false a pessoa permanece na empresa).

A próxima etapa foi separar os dados da tabela entre teste e treino; em seguida importamos o GaussianNB, que possibilita a aplicação do Bayes’ na tabela de teste.

Posteriormente, é feita uma previsão utilizando o algoritmo criado. Nesse momento, é criada a variável previsão, onde são inseridos os números que representam os dados de um funcionário hipotético em cada feature (ou coluna).

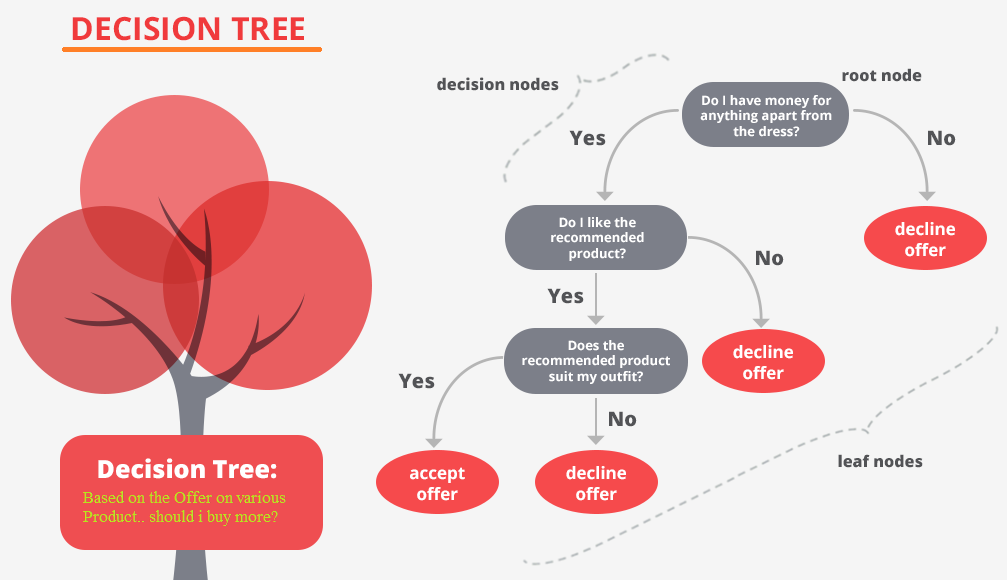
Em seguida, com o mesmo comando utilizado anteriormente (“.predict”), o GaussianNB é aplicado aos atributos previsores da tabela de teste em outra variável criada com esse intuito. Esta etapa é importante para que, importando do sklearn.metrics os algoritmos necessários, os métodos Ensemble sejam aplicados ao modelo.

**4.4.3 Árvore de decisão**

Dentre todos os algoritmos de machine learning, um que se destaca pela sua estrutura visual, é a árvore de decisão. A Árvore de decisão é um algoritmo de aprendizado de máquina supervisionado para classificação e para regressão, o algoritmo divide as variáveis em ramos, esses ramos são calculados conforme a entropia ("confusão" dos dados) e ganho de informação de cada variável (baseado na importância da variável), também é aplicado uma hierarquia na relação desses ramos, existe o nó-raiz e o nós-folha sendo o nó-raiz um dos atributos da base de dados e os nós folhas a classe ou valor que será gerado como resposta.

Na ligação dos nós temos a regra do "se-então", ou seja ao chegar em uma variável A, o algoritmo se pergunta acerca de uma condição, caso a condição seja correspondente ele vai para um lado da árvore, caso contrário irá para o outro lado e no próximo nó segue a mesma lógica até o fim da árvore.

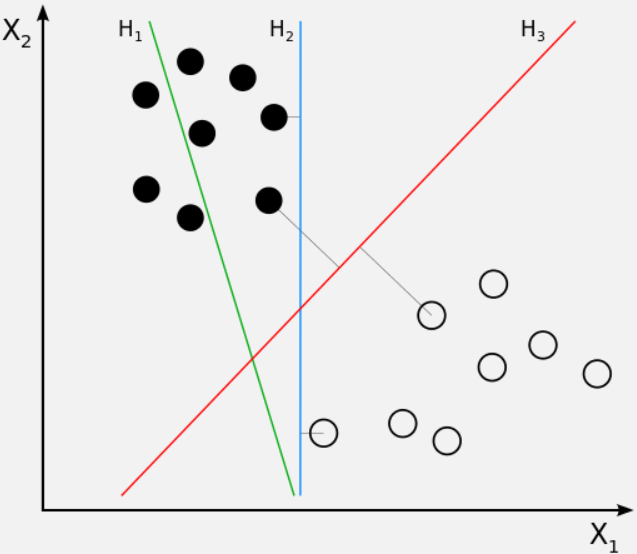
Resumindo é como se o algoritmo aprendesse com a árvore e faz a criação de uma função, que quando novos dados fossem inseridos ele só aplica esses dados na função e gera o resultado, ou seja depende da primeira função para gerar a resposta.



**4.4.4 Support Vector Machine**

Algoritmo SVM: A Support Vector Machines (Máquinas de Vetores de Suporte) é um conjunto de métodos de aprendizado supervisionado usado para classificação, regressão e detecção de *outliers*. Uma SVM constrói um hiperplano ou conjunto deles em um espaço dimensional, que pode ser usado para classificação, regressão ou outras tarefas e distribui os dados de treino de acordo com suas classificações.

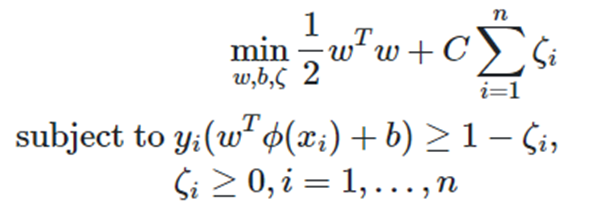
Dentro desse conjunto de métodos, é usado o SVC -Support Vector Classification- que é uma classe capaz de performar várias classificações em um *dataset*. Uma separação ótima é atingida pelo hiperplano com maior distância aos dados de treinamento mais próximos de cada classe (chamados de margens funcionais), sendo todos os dados de um lado do plano classificados de uma maneira e, analogamente, os dados do outro são rotulados com outra classe. A Figura X.X exemplifica a aplicação do algoritmo.



Fonte: Modelos de Predição | SVM.

Algumas vantagens em utilizar esse modelo incluem a adaptabilidade em classificar dados espalhados de maneira não regular, facilidade de aplicação e relativa boa acurácia e bom funcionamento em espaços multidimensionais (com muitas features). Em contrapartida, a interpretabilidade fica mais difícil quanto maior a dimensionalidade dos dados, além de aumentar o tempo para realizar os cálculos.

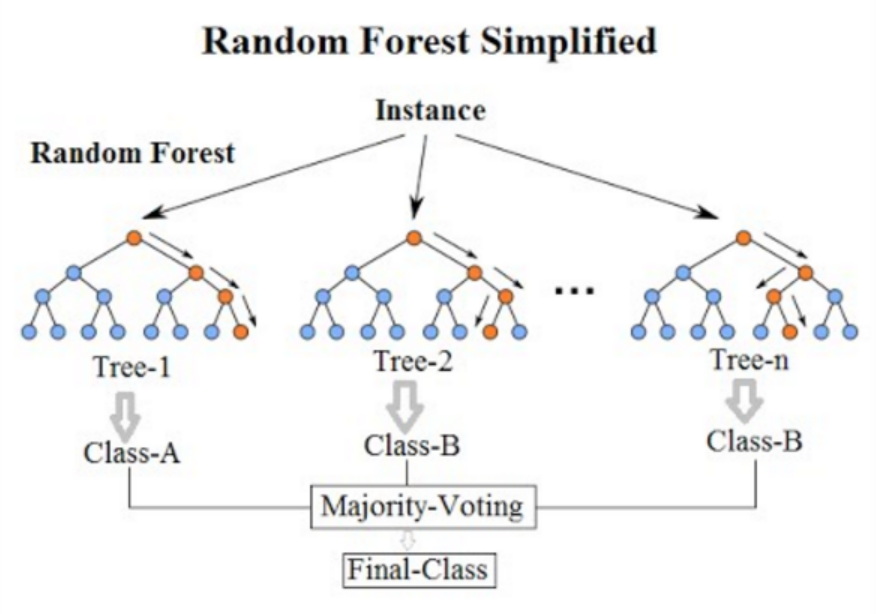
Dados vetores de treino xi ∈ ℝp, i=1,...,n, em duas classes e um vetor y ∈{-1,1}n, a meta é descobrir w ∈ ℝp e b ∈ℝ tal que a previsão dada por *sign(wTΦ(x)+b)* está correta para a maioria das amostras. W e b são, respectivamente, o vetor normal ao hiperplano que é o responsável pelo deslocamento do hiperplano no espaço.

O SVC resolve o seguinte problema primário:

Tentamos então maximizar as margens enquanto aplicamos pequenas punições à amostras classificadas incorretamente ou próximas à margem. Há aceitação de uma certa distância ζi da margem correta. O termo de penalidade C controla a força dessa penalidade e, como resultado, atua como um parâmetro de regularização inverso. Ao final, teremos que a distância entre os dois grupos será a maior possível.

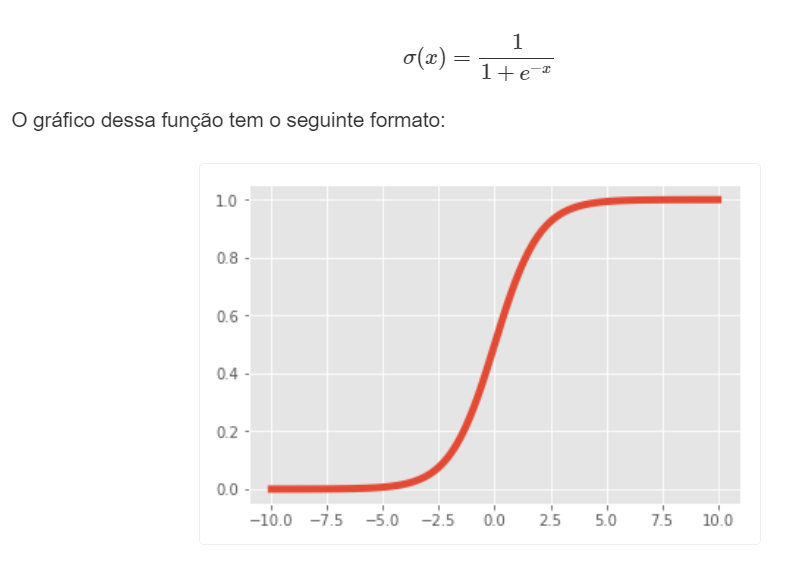
**4.4.5 Random Forest**

O modelo Random Forest é composto de várias árvores de decisão e têm três hiperparâmetros principais, que precisam ser definidos antes do treinamento, tamanho do nó, o número de árvores e o número de recursos amostrados; Logo o modelo compara cada árvore e nos mostra qual delas obteve uma melhor avaliação.



**4.4.6 Regressão Logística**

Algoritmo classificatório de aprendizado supervisionado, nesse caso regressão logística binária, que tem como função categorizar alguma variável por classes. Nesta técnica estatística de mineração de dados, a variável dependente deve ser categórica e as variáveis independentes podem ser métricas ou categóricas. O algoritmo avalia a probabilidade de tal evento ocorrer e entende como as variáveis independentes influenciam em cada evento.

Basicamente, é aplicado a transformação linear para que os valores se tornem probabilidades, então é utilizado a função logística (sigmóide) para criar uma função em ‘S’ no gráfico e classificar as variáveis.

**4.5. Avaliação**

**4.5.1. Features utilizadas**

Para treinar os modelos, foi utilizado um conjunto único de colunas que foram julgadas importantes, sendo elas: "Idade", "NumeroMeses", "Salario Mês", "Cargo", "Estado Civil" "Genero" e "Area", além das criadas depois devido a necessidade (derivação de novos atributos): "Remoto" e "mediaTempoPromoção".

**4.5.2. Separação treino e teste**

Ao desenvolver um modelo de Machine Learning, é necessário separar uma parte dos dados para treino do modelo, e outra para teste. Essa divisão é dada em porcentagem, sendo que no projeto foi usado 70% para teste e 30% para treino para todos os modelos testados.

**4.5.3. Métricas de avaliação**

**4.5.3.1.Acurácia**

A partir dela pode-se saber quantas o modelo acertou dentre as previsões possíveis; é a razão entre o somatório das previsões corretas(verdadeiros positivos com verdadeiros negativos) e o total de previsões.

**4.5.3.2.Recall**

Usando o cálculo de recall sabemos o quão bom o modelo é para prever positivos; é definido como a razão entre os verdadeiros positivos e a soma dos verdadeiros positivos com os falsos negativos.

**4.5.3.3.Precisão**

Mostra qual proporção de identificações positivas foi realmente correta.

**4.5.3.4.Verdadeiro positivo e negativo**

Verdadeiro positivo (true positive - TP) e verdadeiro negativo (true negative - TN) ocorrem quando, no conjunto real, a classe que estamos buscando foi prevista corretamente.

**4.5.3.5.Falso positivo e negativo**

Falso positivo (false positive - FP) e falso negativo (false negative - FN) ocorrem quando, no conjunto real, a classe que estamos buscando prever foi prevista incorretamente.

**4.5.3.6. Hiperparâmetros**

Os hiperparâmetros são parâmetros ajustáveis que permitem controlar e, muitas vezes, melhorar o processo de treinamento do modelo. Por exemplo, com a árvore de decisão, você pode decidir o número de profundidade da árvore e o número de “folhas” em cada ramo. O desempenho do modelo depende muito dos hiperparâmetros. O processo é de localizar a configuração de hiperparâmetros que resultam no melhor desempenho em métricas de avaliação.

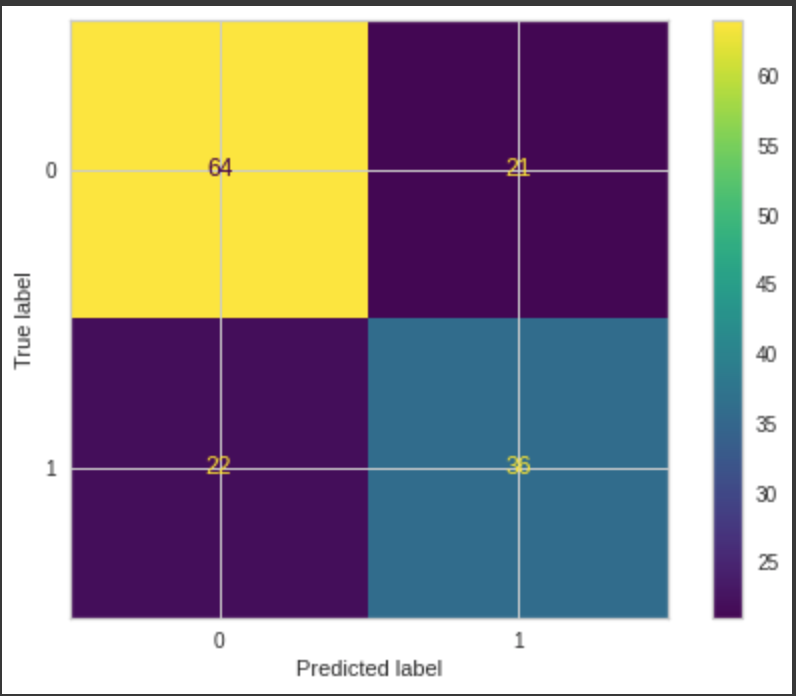
**4.5.4. Resultado das métricas de avaliação**

**4.5.4.1 K Nearest Neighbor**

**4.5.4.1.1 Modelo default**

Ao aplicar o KNN (KNeighborsClassifier()), considerando as Features utilizadas, e os dados separados entre treino e teste, ele gerou um resultado de Acurácia (treino): 72% e de Acurácia (teste): 69%.

|  | Precision | Recall | f1-score | support |
| --- | --- | --- | --- | --- |
| 0 | 0.74 | 0.75 | 0.75 | 85 |
| 1 | 0.63 | 0.62 | 0.63 | 58 |
| Accuracy |  |  | 0.70 | 143 |
| Macro avg | 0.69 | 0.69 | 0.69 | 143 |
| Weighted avg | 0.70 | 0.70 | 0.70 | 143 |



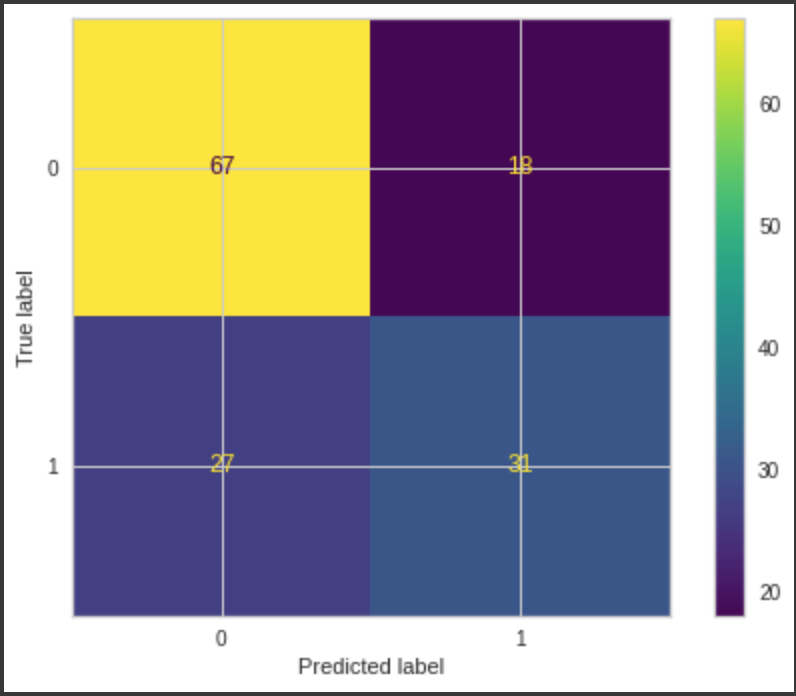
A partir da matriz de confusão do modelo KNN, foi observado que a acuracidade foi mediana, uma vez que dos 58 funcionários que saíram da empresa, o modelo acertou 36 e dos 85 que não saíram da empresa o modelo acertou 64.

**4.5.4.1.2 Modelo com hiperparâmetros**

Após o teste do modelo KNN com parâmetros default, foi estudado a possibilidade de melhorar as métricas de avaliação utilizando hiperparâmetros com métodos de randomização e separação de amostras para múltiplos testes (GridSearchCV e RandomizedSearchCV). No caso do KNN, foi utilizado o RandomSearch, e a melhor combinação foi com os hiperparâmetros: "n\_neighbors", "weights", "algorithm", "leaf\_size", "p", "metric" e com eles, a acurácia obtida foi de: 69%. Como é possível observar, a acurácia permaneceu a mesma comparada ao uso de parâmetros default.

| n\_neighbors | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] |
| --- | --- |
| weights | ['uniform', 'distance'] |
| algorithm | 'auto', 'ball\_tree', 'kd\_tree', 'brute'] |
| leaf\_size | [10, 20, 30, 40, 50, 60, 70, 80, 90, 100] |
| p | [1, 2] |
| metric | ['minkowski', 'euclidean', 'manhattan'] |

|  | Precision | Recall | f1-score | support |
| --- | --- | --- | --- | --- |
| 0 | 0.71 | 0.79 | 0.75 | 85 |
| 1 | 0.63 | 0.53 | 0.58 | 58 |
| Accuracy |  |  | 0.69 | 143 |
| Macro avg | 0.67 | 0.66 | 0.66 | 143 |
| Weighted avg | 0.68 | 0.69 | 0.68 | 143 |



Após a aplicação de hiperparâmetros, é possível perceber uma pequena melhora, pois de 85 pessoas que não saíram, ele acertou 67, e de 58 colaboradores que saíram, o modelo acertou 31.

**4.5.4.1.3 Variância de erro do modelo**

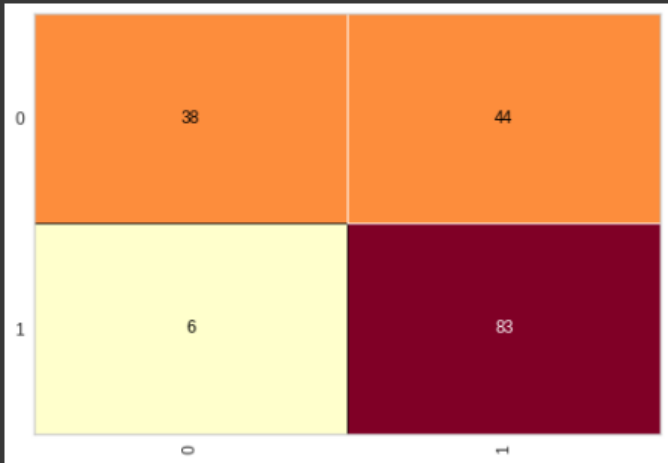
No treinamento do modelo, é normalmente pedido um parâmetro chamado "random\_state", ou seja, qual amostra o modelo que irá usar para treinar. Cada número representa uma amostra diferente, e, testando um intervalo entre esses valores, podemos perceber qual a média de erro do nosso modelo, que no caso do KNN é de 9%

**4.5.4.2 Naive Bayes**

As variáveis utilizadas no algoritmo foram cargo, estado civil, escolaridade e área. Foi feita uma divisão entre teste e treino, então foi aplicado o GaussianNB (importado do “sklearn.naive\_bayes”) aos atributos previsores da tabela de teste. Os resultados foram guardados em uma variável com o nome de previsões e foram comparados aos dados reais do atributo alvo para gerar a acurácia, a matriz de confusão, a precisão, o recall e outras métricas de avaliação.

O resultado da acurácia foi: 70%

As imagens abaixo ajudam a compreender melhor os resultados:



Matriz de confusão do modelo Bayes

Observando a matriz de confusão do modelo de Bayes, pode-se dizer que a acurácia foi mediana, uma vez que ele obteve 38 acertos de 53 previsões das pessoas que continuam na empresa e 83 acertos de 90 previsões das pessoas que saíram da empresa, ou seja, 06 foram falsos negativos e 44 foram falsos positivos.

|  | Precision | Recall | f1-score | support |
| --- | --- | --- | --- | --- |
| 0 | 0.86 | 0.46 | 0.60 | 82 |
| 1 | 0.65 | 0.93 | 0.77 | 89 |
| Accuracy |  |  | 0.71 | 171 |
| Macro avg | 0.76 | 0.70 | 0.69 | 171 |
| Weighted avg | 0.75 | 0.71 | 0.79 | 171 |

imagem ?: resultados das métricas de avaliação do Bayes

**4.5.4.2.1 Aplicação e definição dos hiperparâmetros**

Após o teste do modelo de Bayes com parâmetros default, foi estudado a possibilidade de melhorar as métricas de avaliação utilizando hiperparâmetros com métodos de randomização e separação de amostras para múltiplos testes (GridSearchCV e RandomizedSearchCV). Foram feitos diversos testes, mas o melhor resultado que tivemos foi utilizando: var\_smoothing: 0.000000007935,0.000000008,0.0000000085,0.000000009,0.0000000009,0.00000000009, e a acurácia obtida foi de: 67%. Como é possível observar, a acurácia diminuiu comparada ao uso de parâmetros default.

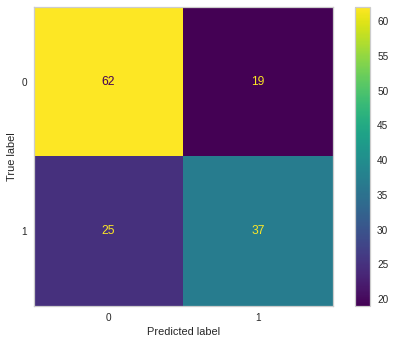
**4.5.4.3 Árvore de decisão**

**4.5.4.3.1 Aplicação do modelo**

A primeira aplicação do algoritmo foi feita somente com o modelo de forma padrão, sem parâmetros e apenas com as features selecionadas. Ele gerou os seguintes resultados:

| Acuracidade (treino) | 1.0 |
| --- | --- |
| Acuracidade (teste) | 0.6923076923076923 |

Para analisar de uma forma mais visual também foi gerado a matriz de confusão, isso nos gera uma análise mais específica de acertos dos funcionários que saíram e que não saíram.





Foi observado que a acuracidade do modelo da árvore de decisão foi relativamente alta já que de 62 funcionários que saíram da empresa o modelo acertou 37 e de 81 que não saíram da empresa o modelo acertou 62, ou seja proporcionalmente a árvore de decisão acertou a grande maioria dos conjuntos de dados.

**4.5.4.3.2 Aplicação e definição dos hiperparâmetros**

Após o teste do modelo árvore de decisão com parâmetros default, foi estudado a possibilidade de melhorar as métricas de avaliação utilizando hiperparâmetros com métodos de randomização e separação de amostras para múltiplos testes (GridSearchCV e RandomizedSearchCV). Os hiperparâmetros estão definidos abaixo:

* GridSearch

| **Parâmetros** | **Valor** |
| --- | --- |
| Criterion | Entropy |
| Max\_depth | 3 |
| Min\_samples\_leaf | 1 |
| Min\_samples\_split | 7 |
| Acuracidade: | 0.7196291270918136 |

* Random Search

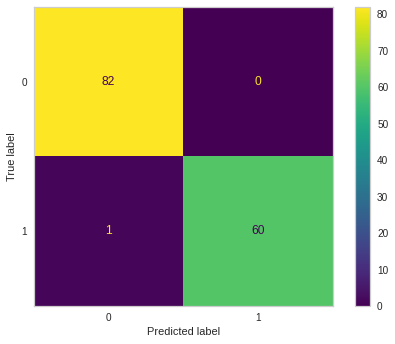
| **Parâmetros** | **Valor** |
| --- | --- |
| Criterion | Gini |
| Max\_depth | 3 |
| Min\_samples\_leaf | 1 |
| Min\_samples\_split | 7 |
| Acuracidade: | 0.7015829941203074 |

**4.5.4.3.3 Variância de erro do modelo**

No treinamento do modelo, é pedido um parâmetro chamado "random\_state". Foi testado números em um intervalo de 42 a 100 cada número representa uma amostra diferente, e, testando um intervalo entre esses valores, podemos perceber qual a média de erro do nosso modelo, que no caso do Árvore de decisão é de 10%. Segue um exemplo de 5 intervalos abaixo:

| Acuracidade (teste): 70 | 0.9230769230769231 |
| --- | --- |
| Acuracidade (teste): 71 | 0.8881118881118881 |
| Acuracidade (teste): 72 | 0.986013986013986 |
| Acuracidade (teste): 73 | 0.9090909090909091 |
| Acuracidade (teste): 74 | 0.9370629370629371 |

De acordo com o intervalo apresentado anteriormente podemos concluir que a amostra que mais nos dá a maior classificação é o de 72 com uma acuracidade de 98%.

****

**4.5.4.4 Support Vector Machine**

Para a aplicação do SVM, foram usados como atributos de análise os seguintes hiperparâmetros: idade do colaborador, Número de meses na empresa, salário/mês do colaborador, seu cargo, gênero, estado civil, área na qual trabalha, se trabalha presencial ou remotamente e a média de tempo entre as promoções. Após a escolha desses elementos, separamos amostras de treinamento e teste na proporção 70%-30% randomicamente, preparamos o modelo com os devidos dados e verificamos os resultados retornados. Por fim, executando o modelo com diferentes configurações de amostras, obtivemos uma acurácia média de 65% para os dados de teste e pontuações de métricas mostradas abaixo.

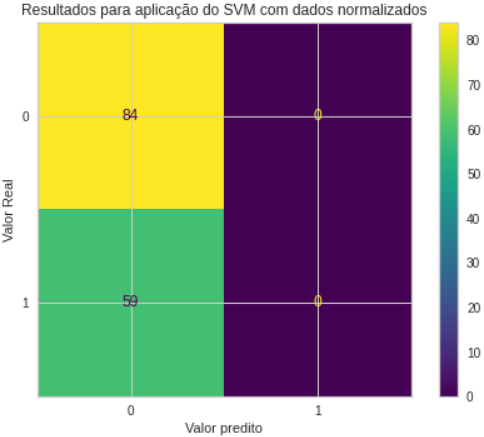
|  | Precision | Recall | f1-score | support |
| --- | --- | --- | --- | --- |
| 0 | 0.66 | 0.95 | 0.78 | 84 |
| 1 | 0.81 | 0.29 | 0.43 | 59 |
| Accuracy |  |  | 0.68 | 143 |
| Macro avg | 0.73 | 0.62 | 0.60 | 143 |
| Weighted avg | 0.72 | 0.68 | 0.63 | 143 |

****

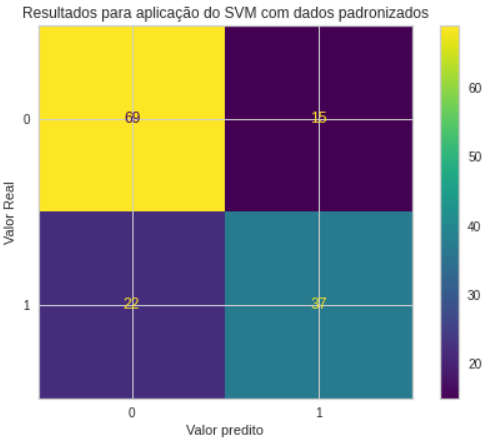
Pode-se perceber que a acuracidade do modelo SVM foi baixa, considerando que de 59 funcionários que saíram da empresa o modelo acertou 17 e de 84 que não saíram da empresa o modelo acertou 80, ou seja proporcionalmente,o SVM acertou a grande maioria dos conjuntos de dados dos colaboradores que permanecem, mas erra com frequência na classificação da saída.

Após o teste padrão, foram utilizadas as técnicas de padronização e normalização dos dados, a fim de descobrir o impacto dessas transformações no modelo, tanto separadamente quanto simultaneamente. Analisando os resultados, percebemos uma diminuição na acurácia média quando aplicada a normalização, um aumento brando quando aplicada a padronização; quando simultaneamente aplicadas, o resultado é idêntico à última transformação feita, ou seja, se os dados foram tratados na ordem padronização->normalização, o resultado obtido será de 61%, caso contrário, a acurácia será de 67%.

|  | Precision | Recall | f1-score | support |
| --- | --- | --- | --- | --- |
| 0 | 0.59 | 1.00 | 0.74 | 84 |
| 1 | 0.00 | 0.00 | 0.00 | 59 |
| Accuracy |  |  | 0.59 | 143 |
| Macro avg | 0.29 | 0.50 | 0.37 | 143 |
| Weighted avg | 0.35 | 0.59 | 0.43 | 143 |



|  | Precision | Recall | f1-score | support |
| --- | --- | --- | --- | --- |
| 0 | 0.76 | 0.82 | 0.79 | 84 |
| 1 | 0.71 | 0.63 | 0.67 | 59 |
| Accuracy |  |  | 0.74 | 143 |
| Macro avg | 0.73 | 0.72 | 0.73 | 143 |
| Weighted avg | 0.74 | 0.74 | 0.74 | 143 |



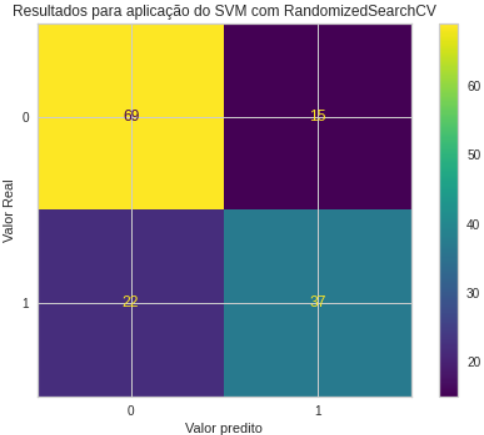
Após esses resultados, aplicamos métodos para otimização de escolha de hiperparâmetros utilizando métodos de randomização e separação de amostras para múltiplos testes (GridSearchCV e RandomizedSearchCV). Para testar a aplicação dessas técnicas, rodamos 48 vezes com amostras diferentes para avaliar a média entre as acurácias.

Para o GridSearchCV, a acurácia para dados de treino aumentou de 62% para 68%; a acurácia dos dados de teste aumentou de 65% para 70%.

|  | Precision | Recall | f1-score | support |
| --- | --- | --- | --- | --- |
| 0 | 0.76 | 0.82 | 0.79 | 84 |
| 1 | 0.71 | 0.63 | 0.67 | 59 |
| Accuracy |  |  | 0.74 | 143 |
| Macro avg | 0.73 | 0.72 | 0.73 | 143 |
| Weighted avg | 0.74 | 0.74 | 0.74 | 143 |

Para o RandomizedSearchCV, a acurácia média com 48 testes para os dados de teste subiu de 65% para 70%.

|  | Precision | Recall | f1-score | support |
| --- | --- | --- | --- | --- |
| 0 | 0.76 | 0.82 | 0.79 | 84 |
| 1 | 0.71 | 0.63 | 0.67 | 59 |
| Accuracy |  |  | 0.74 | 143 |
| Macro avg | 0.73 | 0.72 | 0.73 | 143 |
| Weighted avg | 0.74 | 0.74 | 0.74 | 143 |

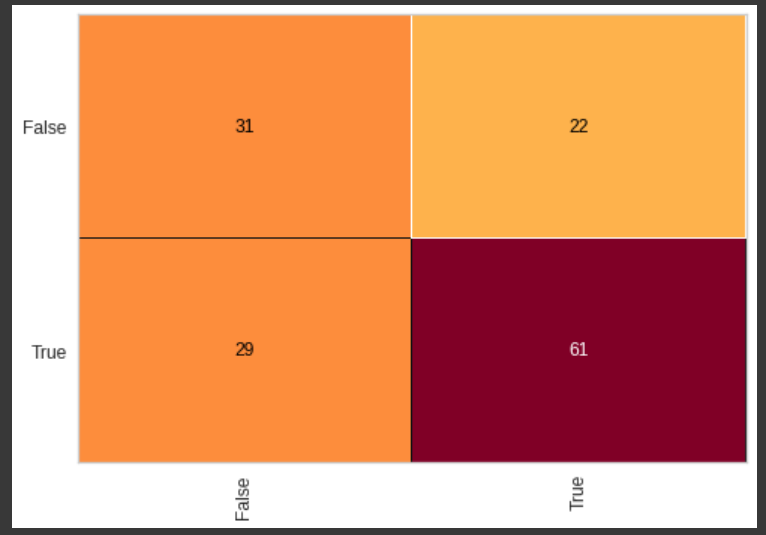


Após essas análises, pode-se concluir que a procura de otimização de parâmetros aumentou timidamente a acurácia dos resultados.

**4.5.4.5 Random Forest**

Antes de aplicar o modelo separamos os dados em “para teste” e “para treino”, e importamos a biblioteca sklearn.ensemble para conseguirmos usar a função de Random forest. Depois, já na função aplicada, definimos a quantidade de “árvores” a serem criadas (usamos 40) e determinamos outras variáveis necessárias para o código.

Por fim os resultados foram guardados em uma variável e foram comparados aos dados reais do atributo alvo para gerar a acurácia, a matriz de confusão, a precisão, o recall e outras métricas de avaliação.



Matriz de confusão do modelo Random Forest

Observando a matriz de confusão do modelo Random Forest, pode-se dizer que a acurácia foi mediana, uma vez que obteve 31 das 53 previsões de permanência na empresa e 61 das 90 de saída da empresa estão corretas, ou seja 29 falsos negativos e 22 falsos positivos.

|  | Precision | Recall | f1-score | support |
| --- | --- | --- | --- | --- |
| 0 | 0.52 | 0.58 | 0.55 | 53 |
| 1 | 0.73 | 0.68 | 0.71 | 90 |
| Accuracy |  |  | 0.64 | 143 |
| Macro avg | 0.63 | 0.63 | 0.63 | 143 |
| Weighted avg | 0.65 | 0.64 | 0.65 | 143 |

imagem ?: resultados das métricas de avaliação do Random Forest

| Acuracidade (teste) | 0.6433566433566433 |
| --- | --- |

**4.5.4.4.1 Aplicação e definição dos hiperparâmetros**

Após o teste do modelo Random Forest com parâmetros default, foi estudado a possibilidade de melhorar as métricas de avaliação utilizando hiperparâmetros com métodos de randomização e separação de amostras para múltiplos testes (GridSearchCV e RandomizedSearchCV). Foram feitos diversos testes, mas o melhor resultado que tivemos foi utilizando: 'n\_estimators': [1, 2, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90], 'min\_samples\_split': [2, 5, 10], 'min\_samples\_leaf': [1, 2, 4], 'max\_features': ['auto', 'sqrt'], max\_depth': [1, 2, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90], 'max\_depth': [1, 2, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90], 'bootstrap': [True, False] e a acurácia obtida foi de: 77%. Como é possível observar, a acurácia aumentou comparada ao uso de parâmetros default.

**4.5.4.6 Regressão Logística**

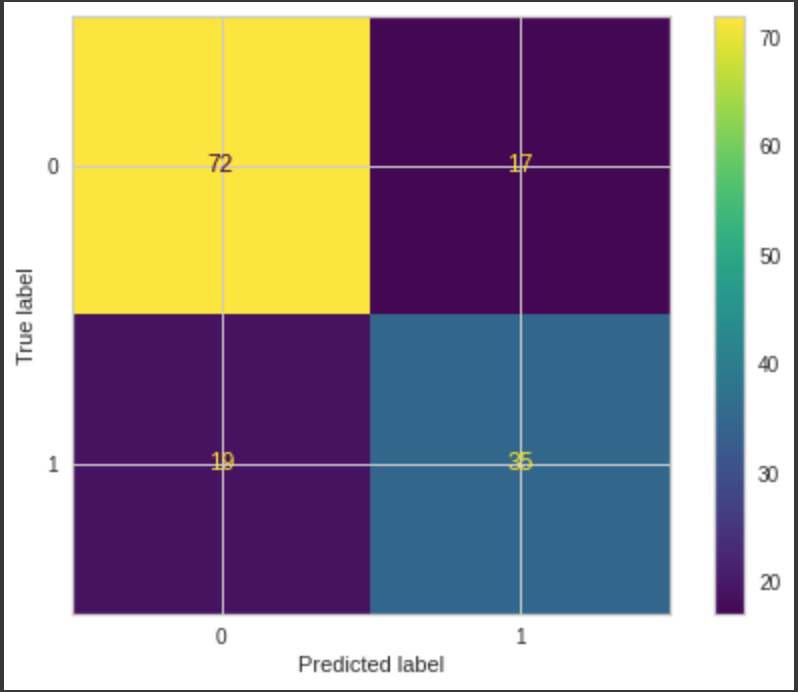
Antes de aplicar o algoritmo é necessário separar os dados em “para teste” e “para treino” e definir o "random state”. A divisão dos dados utilizada foi 70% para treino, 30% para teste e o random state igual 6.

Depois, foi utilizado a função MinMaxScaler() para normalizar os atributos, transformando os dados para que fiquem em um intervalo entre 0 e 1, para que não haja valores extremos.

Importando a biblioteca sklearn.linear\_model para rodar o algoritmo da regressão logística.

Por fim, após Importar a biblioteca sklearn.linear\_model para rodar o algoritmo da regressão logística. os resultados foram comparados aos dados reais do atributo alvo para gerar a acurácia, a matriz de confusão, a precisão, o recall e outras métricas de avaliação.

**Matriz de confusão:**

****

|  | Precision | Recall | f1-score | support |
| --- | --- | --- | --- | --- |
| 0 | 0.79 | 0.81 | 0.80 | 89 |
| 1 | 0.67 | 0.65 | 0.66 | 54 |
| Accuracy |  |  | 0.75 | 143 |
| Macro avg | 0.73 | 0.73 | 0.73 | 143 |
| Weighted avg | 0.75 | 0.75 | 0.75 | 143 |

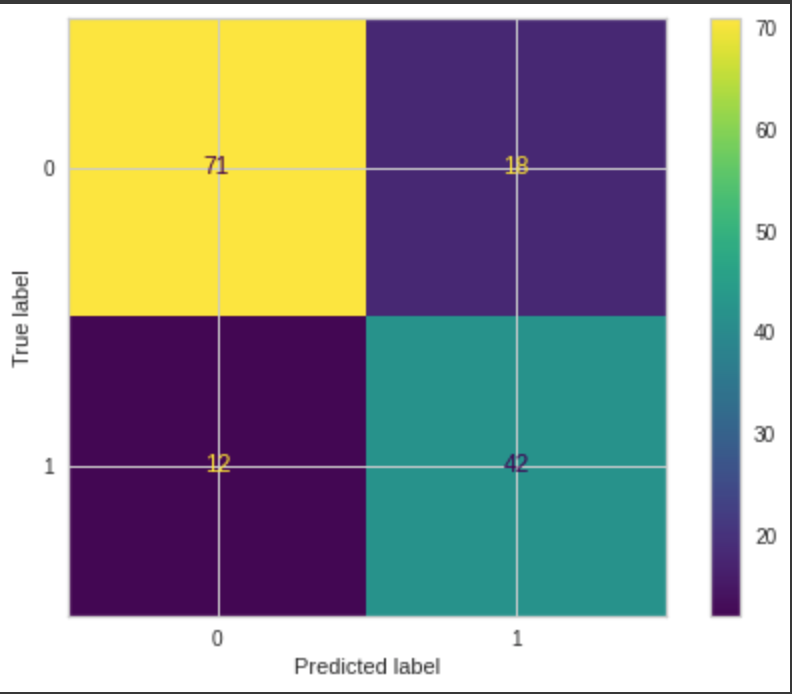
A matriz de confusão da regressão logística mostra que o algoritmo previu corretamente 72 dos 89 que não saíram e 35 dos 54 que saíram.

**4.5.4.6.1 Aplicação e definição dos hiperparâmetros**

Depois de analisar os resultados, foi aplicado a otimização dos hiperparâmetros, utilizando os métodos GridSearch e Randomized Search para identificar os melhores parâmetros.

| C | 100 |
| --- | --- |
| penalty | l1 |
| solver | saga |

**Matriz de confusão com os hiperparâmetros:**

****

|  | Precision | Recall | f1-score | support |
| --- | --- | --- | --- | --- |
| 0 | 0.86 | 0.80 | 0.83 | 89 |
| 1 | 0.70 | 0.78 | 0.74 | 54 |
| Accuracy |  |  | 0.79 | 143 |
| Macro avg | 0.78 | 0.79 | 0.78 | 143 |
| Weighted avg | 0.80 | 0.79 | 0.79 | 143 |

A matriz de confusão da regressão logística com a aplicação dos hiperparâmetros mostra que o algoritmo previu corretamente 71 dos 89 que não saíram e 42 dos 54 que saíram, mostrando um aumento de 5% na acurácia.

**4.5.4.6.2 Variância do modelo**

Após rodar 10 vezes o algoritmo com diferentes amostras do banco de dados, a acurácia variou 10%.

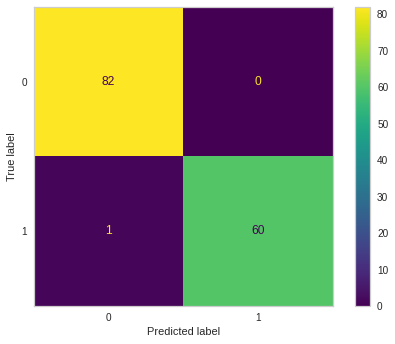
**4.5.5. Conclusão dos modelos**

Analisando os resultados de todos os seis modelos, chegou-se à conclusão de que o modelo de maior qualidade é o da árvore de decisão, uma vez que apresenta os maiores índices de avaliação.

Após testar muitas vezes o mesmo modelo com vários atributos diferentes, chegamos no resultado final de 98% de acurácia utilizando os melhores parâmetros encontrados para o algoritmo de árvore de decisão e na divisão de treino e teste.

Segue abaixo será apresentado a acurácia de treino e teste junto com a matriz de confusão:

| Acuracidade (treino) | 0.8734939759036144 |
| --- | --- |
| Acuracidade (teste) | 0.986013986013986 |



A partir disso, é possível relacionar os resultados obtidos com as análises de mercado citadas no índice 4.1. A partir do Value Proposition Canvas, percebe-se que a empresa do cliente possui alguns fatores que aumentam a taxa de turnover, e o nosso projeto vem com o intuito de ajudá-lo a melhorar essa taxa, uma vez que, a partir da resposta do modelo, ele poderá investigar determinado funcionário e tomar as providências necessárias para impedir que ele saia.

Por fim, a hipótese inicial do grupo era na questão salarial, uma vez que esse atributo foi usado na solução final, e sem ele a acurácia diminui para 72%. Além disso, outra hipótese que tínhamos era em relação a idade. De acordo com uma pesquisa feita pela revista Veja em 2018, pessoas mais velhas costumam se arriscar menos no mercado, permanecendo mais tempo na empresa que entram, ao contrário de pessoas mais jovens, que costumam receber mais ofertas de emprego, e por isso se sentem mais seguras em trocar de trabalho.

**5. Conclusões e Recomendações**

Escreva, de forma resumida, sobre os principais resultados do seu projeto e faça recomendações formais ao seu parceiro de negócios em relação ao uso desse modelo. Você pode aproveitar este espaço para comentar sobre possíveis materiais extras, como um manual de usuário mais detalhado na seção “Anexos”.

Não se esqueça também das pessoas que serão potencialmente afetadas pelas decisões do modelo preditivo, e elabore recomendações que ajudem seu parceiro a tratá-las de maneira estratégica e ética.

**6. Referências**

Nesta seção você deve incluir as principais referências de seu projeto, para que seu parceiro possa consultar caso ele se interessar em aprofundar.

Utilize a norma ABNT NBR 6023 para regras específicas de referências. Um exemplo de referência de livro:

SOBRENOME, Nome. **Título do livro**: subtítulo do livro. Edição. Cidade de publicação: Nome da editora, Ano de publicação.

CRISP-DM Help Overview. [*S. l.*]: © Copyright IBM Corporation, ca. 2021. Disponível em: https://www.ibm.com/docs/en/spss-modeler/saas?topic=dm-crisp-help-overview. Acesso em: 8 set. 2022.

COLABORATORY: Perguntas frequentes. [*S. l.*]: Google, ca. 2022. Disponível em: https://research.google.com/colaboratory/intl/pt-BR/faq.html. Acesso em: 8 set. 2022.

O QUE é GitHub e Como Usá-lo. [*S. l.*]: © 2004-2022 hostinger.com.br - Hospedagem de Sites, Cloud e VPS premium e Serviços de Registro de Domínio., 31 jan. 2022. Disponível em: https://www.hostinger.com.br/tutoriais/o-que-github. Acesso em: 8 set. 2022.

HOTZ, NICK. What is CRISP DM?. [*S. l.*]: Copyright 2022 @ Data Science Process Alliance. All rights reserved., 2022. Disponível em: https://www.datascience-pm.com/crisp-dm-2/. Acesso em: 8 set. 2022.

COUTINHO, Bernardo. Modelos de Predição | SVM: Aprenda a criar seu primeiro algoritmo de classificação com SVM.. [*S. l.*], 28 jul. 2019. Disponível em: https://medium.com/turing-talks/turing-talks-12-classificação-por-svm-f4598094a3f1. Acesso em: 9 set. 2022.

1.4. Support Vector Machines. Scikit-learn 1.1.2. [*S. l.*]: © 2007 - 2022, scikit-learn developers (BSD License)., entre 2015 e 2022. Disponível em: https://scikit-learn.org/stable/modules/svm.html#svm-mathematical-formulation. Acesso em: 9 set. 2022.

[https://medium.com/turing-talks/turing-talks-12-classificação-por-svm-f4598094a3f1](https://medium.com/turing-talks/turing-talks-12-classifica%C3%A7%C3%A3o-por-svm-f4598094a3f1)

[Salesforce quer mais parceiros | Notícias | Baguete](https://www.baguete.com.br/noticias/27/06/2022/salesforce-quer-mais-parceiros)

[Novo estudo sobre mercado Salesforce reforça avanço do movimento Low-Code/No Code | iMasters](https://imasters.com.br/noticia/novo-estudo-sobre-mercado-salesforce-reforca-avanco-do-movimento-low-code-no-code)

[Tecnologia em nuvem: veja as tendências para 2022!](https://santodigital.com.br/5-tendencias-de-tecnologia-em-nuvem-para-2021/)

**Anexos**

Utilize esta seção para anexar materiais como manuais de usuário, documentos complementares que ficaram grandes e não couberam no corpo do texto etc.