



Inteli

# FireData Everymind



# 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.

			Finalização das mudanças propostas na versão 4.0
09/09/2022	Alexandre Fonseca, Bruno Meira, Gabriela Morais, João Alcaraz, Filipi Kikuchi, Lucas Pereira e Sofia Pimazzoni	4.1	
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
05/10/2022	Alexandre Fonseca, Bruno Meira, Gabriela Morais, João Alcaraz, Filipi Kikuchi, Lucas Pereira e Sofia Pimazzoni	5.1	Atualização dos resultados e revisão dos índices no sumário

# Sumário

<b>Sumário</b>	<b>4</b>
<b>1. Introdução</b>	<b>7</b>
<b>2. Objetivos e Justificativa</b>	<b>8</b>
2.1. Objetivo Geral	8
2.2 Objetivos Específicos	8
2.3 Justificativa	9
<b>3. Metodologia</b>	<b>10</b>
3.1. CRISP-DM	10
3.1.1. Fases do CRISP-DM	10
3.2. Ferramentas	11
3.3. Principais técnicas empregadas	11
<b>4. Desenvolvimento e Resultados</b>	<b>12</b>
4.1. Compreensão do Problema	12
4.1.1. Contexto da indústria	12
4.1.2. Análise SWOT	13
4.1.3. Planejamento Geral da Solução	14
4.1.4. Value Proposition Canvas	14
4.1.5. Matriz de Riscos	15
4.1.6. Personas	16
4.1.7. Jornadas do Usuário	17
<b>4.2. Compreensão dos Dados</b>	<b>19</b>
4.2.1 Descrição dos dados	19
4.2.1.1 Descrição do agrupamento e mescla	22
4.2.1.2 Descrição dos riscos e contingências	22
4.2.1.3 Descrição dos criterios de escolha para análises iniciais	22
4.2.1.4 Descrição das restrições de segurança	22
4.2.2 Descrição estatística básica dos dados	23

4.2.3 Descrição da predição desejada	27
<b>4.3. Preparação dos Dados</b>	<b>27</b>
4.3.1 Classificação e formatação de dados para modelagem	30
4.3.2 Remover ou substituir registros em branco	31
4.3.3 Seleção de um subconjunto de amostras para análise	32
4.3.4 Derivação de novos atributos	32
4.3.5 Colunas não utilizadas	33
4.3.6 Mesclar conjuntos de dados e registros	33
4.3.7 Oversampling	34
4.3.8 Normalização	34
4.3.9 Padronização	35
<b>4.4. Modelagem</b>	<b>35</b>
4.4.1 K-Nearest-Neighbor	35
4.4.2 Naïve Bayes	36
4.4.3 Árvore de decisão	37
4.4.4 Random Forest	38
4.4.5 Support Vector Machine	39
4.4.6 Regressão Logística	40
4.4.7 Justificativa das escolhas dos algoritmos	40
<b>4.5. Avaliação</b>	<b>41</b>
4.5.1. Features utilizadas	41
4.5.2. Separação treino e teste	41
4.5.3. Validação cruzada	41
4.5.4. Métricas de avaliação	42
4.5.4.1. Acurácia	42
4.5.4.2. Recall	43
4.5.4.3. Precisão	43
4.5.4.4. Verdadeiro positivo e negativo	43
4.5.4.5. Falso positivo e negativo	43
4.5.4.6. Hiperparâmetros	43
4.5.4.7. Curva ROC	43

4.5.4.8. Matriz de confusão	44
4.5.5. Resultado das métricas de avaliação	45
4.5.5.1 K Nearest Neighbor	45
4.5.5.1.1 Modelo default	45
4.5.5.1.2 Modelo com hiperparâmetros	46
4.5.5.1.3 Variância de erro do modelo	47
4.5.5.2 Naïve Bayes	47
4.5.5.2.1 Modelo default	47
4.5.5.2.2 Aplicação e definição dos hiperparâmetros	49
4.5.5.2.3 Variância de erro do modelo	50
4.5.5.3 Árvore de decisão	50
4.5.5.3.1 Modelo default	50
4.5.5.3.2 Aplicação e definição dos hiperparâmetros	51
4.5.5.3.3 Variância de erro do modelo	52
4.5.5.4 Support Vector Machine	53
4.5.5.4.1 Modelo default	53
4.5.5.4.2 Aplicação e definição dos hiperparâmetros	54
4.5.5.5 Random Forest	59
4.5.5.5.1 Modelo default	59
4.5.5.5.2 Aplicação e definição dos hiperparâmetros	60
4.5.5.5.3 Variância de erro do modelo	61
4.5.5.6 Regressão Logística	61
4.5.5.6.1 Modelo default	61
4.5.5.6.2 Aplicação e definição dos hiperparâmetros	62
4.5.5.6.3 Variância do modelo	63
4.5.6. Conclusões dos modelos	64
4.5.6.1 Curva ROC	64
<b>5. Conclusões e Recomendações</b>	<b>67</b>
<b>6. Referências</b>	<b>68</b>
<b>Anexos</b>	<b>69</b>

# 1. Introdução

A Everymind é uma empresa de consultoria especializada em ERP, software de gestão empresarial, da Salesforce, oferecendo soluções personalizadas para o cliente. A empresa oferece diversos serviços de Cloud, como Sales Cloud(CRM), Service Cloud, Marketing Cloud, entre outros e atende clientes em diferentes etapas do ciclo de vida de uma empresa.

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).

Embora a empresa tenha crescido nos últimos anos, o problema trazido pela empresa expõe sua preocupação com a taxa de Turnover, principalmente na área dos Devs, e espera entender os fatores que mais contribuem para a rotatividade de funcionários. Dessa forma, a Everymind propõe a criação de um modelo preditivo que indique possíveis tendências de saída.

## 2. Objetivos e Justificativa

### 2.1. Objetivo 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 Objetivos Específicos

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
- Naïve 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. Para descrever a metodologia neste documento, utilizamos como base o artigo “CRISP-DM Help Overview” da IBM.

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.

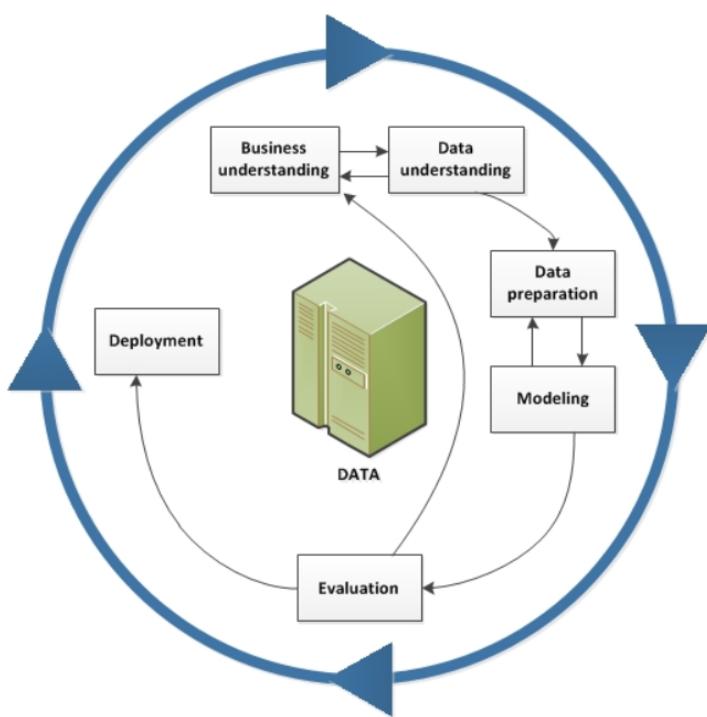


Figura 1. Ciclo de vida do processo de mineração de dados

#### 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, Naïve-Bayes, Árvore de Decisão, Support Vector Machine, 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 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.

No contexto da indústria, a Everymind é uma empresa que oferece sistemas ERP especializados em Salesforce (Enterprise Resource Planning - Planejamento de Recursos Empresariais, em tradução direta), que são soluções tecnológicas que interligam os dados e processos de uma empresa. Por conta da alta demanda por esse tipo de serviço e procura por desenvolvedores, o setor vem enfrentando altas taxas de rotatividade nas empresas (Turnover - mede o número de saída de funcionários de uma empresa em um período de tempo), o que pode ser ocasionado por vários fatores.

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 alojar 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)
- Empresas que oferecem serviços e sistemas ERP (NetSuite, monday projects, 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.

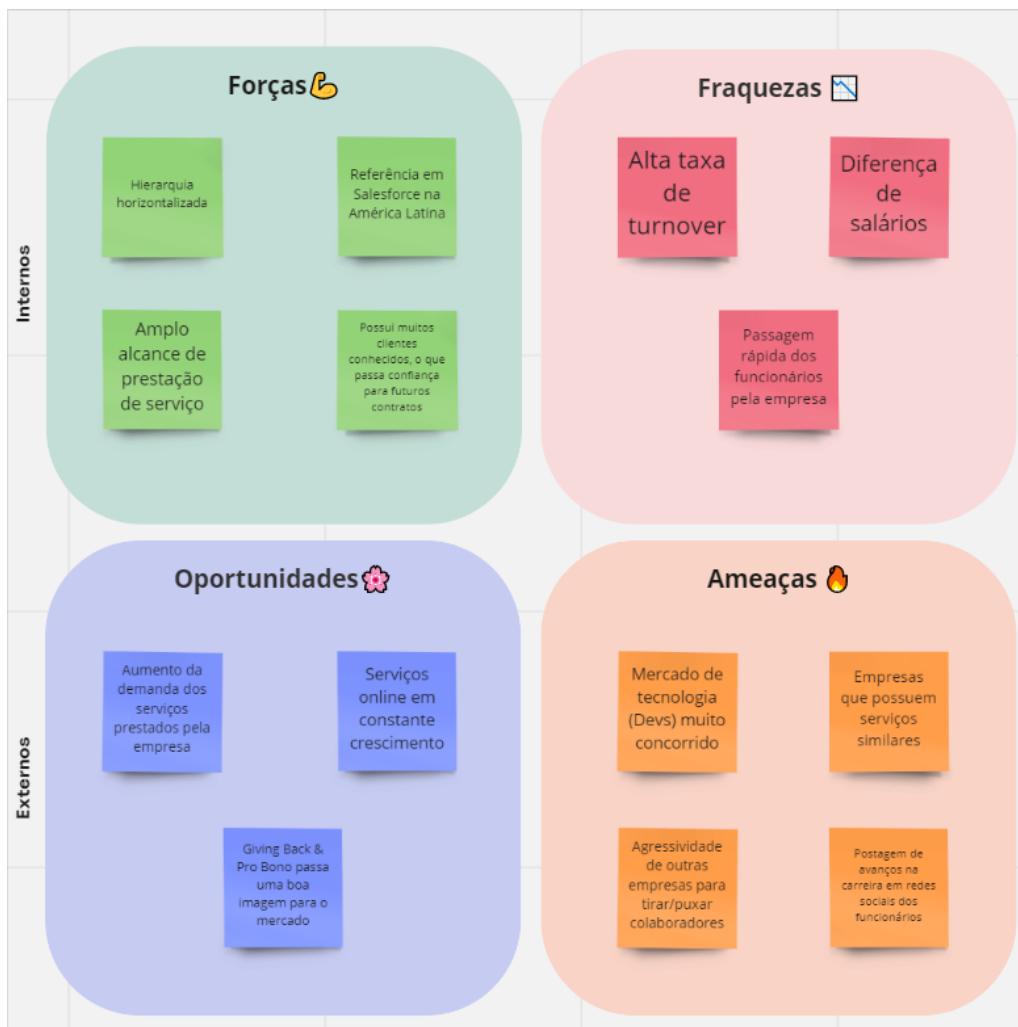


Figura 2. Análise SWOT do cenário onde a solução está inserida

### 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 Recursos Humanos, facilitando os processos de gestão, 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. O uso do modelo pode apontar os fatores que favorecem a permanência de um colaborador e evidenciar possíveis padrões nos dados com certo grau de confiabilidade. 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.

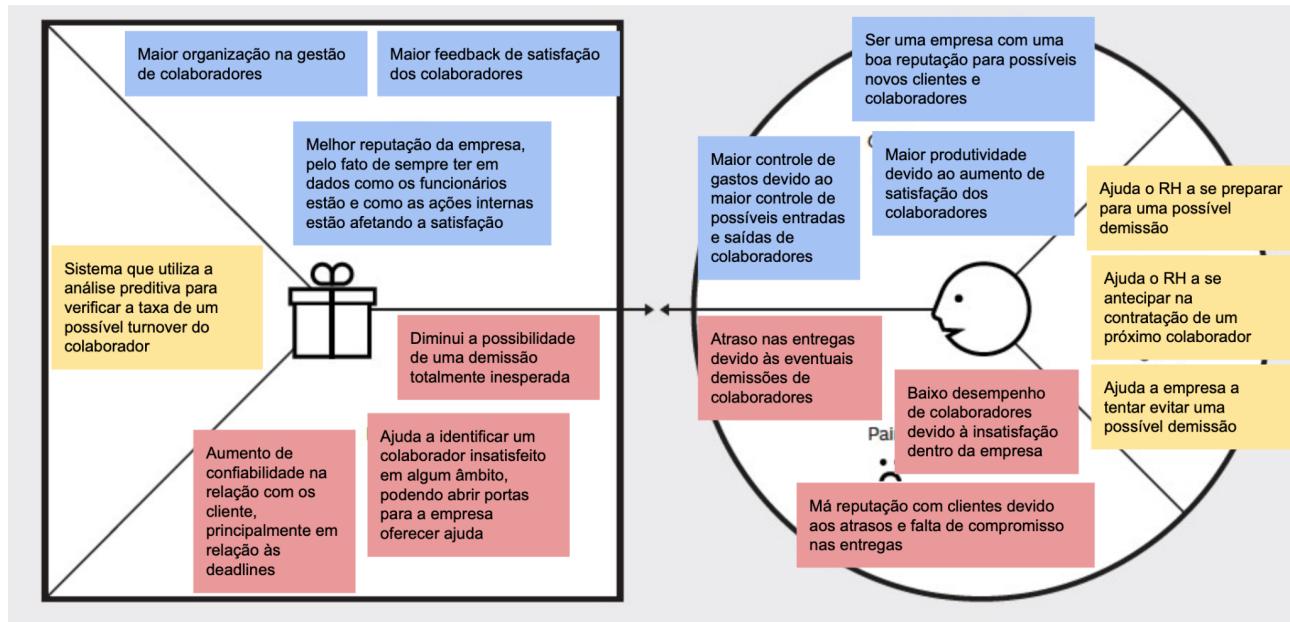


Figura 3. Proposta de valor dos clientes da solução proposta

#### 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

Probabilidade		Ameaças						Oportunidades					
Muito Alto	5	-	13	10	-	-	-	-	-	-	-	-	-
Alto	4	-	11	-	-	-	8	6	-	-	-	-	-
Médio	3	-	12	1	-	-	7	-	-	-	-	-	-
Baixa	2	-	14	15	2	5	-	-	-	-	-	-	-
Muito Baixa	1	-	-	3	9	4	-	-	-	-	-	-	-
		1	2	3	4	5	5	4	3	2	1		
	Muito Baixa	Baixa	Médio	Alta	Muito Alta	Muito Alta	Alta	Médio	Baixa	Muito Baixa			
Impacto													

Figura 4. Análise e validação dos riscos da solução proposta

#### 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:solução

## Kaique Romano (Colaborador)



- 22 anos
- DEV júnior
- Insatisfeito com o salário
- Sem perspectiva de evolução da empresa
- Recebe muitas ofertas de trabalho
- Com o diagnóstico da predição, seria possível a empresa perceber a insatisfação e, com isso, entrar em contato com o colaborador
- Objetivo: Neymar é um morador de comunidade, a profissão de desenvolvedor abriu muitas portas para ele, com isso, Neymar deseja trabalhar em um ambiente saudável e ter uma boa remuneração para ajudar na renda de sua família.
- Dores: Neymar não ganha o tanto quanto ele gostaria, além de não ter uma perspectiva de evolução, fazendo com que sua rotina seja extremamente maçante, impossibilitando-o de aproveitar mais tempo com sua família. Devido a alta demanda de trabalho, ele desenvolveu algumas crises, como a ansiedade, sendo assim, ele também deseja um ambiente de trabalho mais receptivo.

Figura 5. Análise e descrição da persona Kaique Romano

## Bianca Nepumoceno (time de RH)



- 28 anos
- Faz parte do time de RH
- Workaholic
- Extremamente preocupada com a saúde da empresa
- Está satisfeita com o seu trabalho
- Com o diagnóstico da predição, seria possível fazer o seu trabalho de forma mais eficiente
- Objetivo: Ela cresceu em uma família de classe média e nunca passou dificuldades, sendo assim, ela deseja se tornar a Head de Recursos Humanos, mas não tem pressa em alcançar esse objetivo.
- Dores: Perde muitos funcionários sem saber o motivo, e isso faz com que a empresa fique com uma reputação ruim, pois isso atrasa as entregas para os clientes, e mostra uma instabilidade. Ela deseja entender com mais propriedades quais são as insatisfações do colaborador.

Figura 6. Análise e descrição da persona Bianca Nepumoceno

## Antony Vicente S. Maravalhas (Squad Leader)



- 36 Anos
- Jogador de golfe
- "Love is so short, forgetting is so long"
- É um dos pilares da empresa, consegue afirmar sua presença pelo seu carisma e comprometimento.
- Com o diagnóstico da predição, seria possível fazer o seu trabalho de forma mais eficiente.
- Objetivo: Quer deixar seu legado como líder, guiando sua equipe da melhor maneira possível
- Dores: Muitas vezes sente que seu time está desfalcado. Além disso, sempre entram pessoas novas em seu time devido à alta taxa de turnover, mas ele possui dificuldade para descobrir o motivo disso.

Figura 7. Análise e descrição da persona Antony Vicente

### 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. Foi decidido que era melhor fazer a jornada do usuário apenas dos cargos mais impactados pelo modelo.

 <b>Antony Vicente S. Maravalhas</b>		<b>Expectativas</b> Conseguir engajar novamente o colaborador e reestruturar seu time		
<b>Cenário:</b> Antony está insatisfeito com desempenho um membro de sua equipe, pois ele não está engajado e esta atrasando muitas entregas do projeto atual.				
FASE 1 (Consultar o modelo)	FASE 2 (Entender o problema)	FASE 3 (Analizar o problema)	FASE 4 (Solucionar o problema)	FASE 5 (Retorno do rendimento)
1 - Consultar o algoritmo 2 - O funcionário em questão foi classificado como "propenso a sair"	1. Conversar com o colaborador para entender suas dores 2. O membro está agindo dessa forma pois não sentia que seu esforço estava sendo reconhecido.	1. Analisou a performance do funcionário na entrega dos últimos projetos 2. Concluiu que o funcionário foi o diferencial em diversos projetos bem-sucedidos e, de fato, merecia reconhecimento.	1. Consultando no sistema, descobriu que o funcionário estava com uma promoção pendente há alguns meses. 2. Se convenceu de que o membro de sua equipe realmente merecia a promoção.	1. O funcionário recebe o reconhecimento que estava esperando 2. O engajamento da equipe volta ao normal e as entregas voltam a ser satisfatórias.
<b>Oportunidades</b>		<b>Responsabilidades</b>		
Essa situação mostra como é importante reconhecer seus funcionários quando estão fazendo um bom trabalho para manter o foco da equipe.		Para aprimorar o reconhecimento dos colaboradores, a empresa poderia consultar o modelo preditivo com mais frequência, para ficar alerta de quais pessoas estão propensas a sair e poder tomar as devidas providências.		

Figura 8. Análise e descrição do cenário e jornada do Antony Vicente

 <b>Kaique Romano</b>		<b>Expectativas</b> Espera conseguir o reconhecimento que deseja, pois já está acostumado com a vida e a cultura da empresa, e não deseja sair		
<b>Cenário:</b> Kaique está insatisfeito com a falta de reconhecimento da empresa e está considerando sair já que recebe muitas ofertas de trabalho				
FASE 1 (Comentar com um amigo suas impressões atuais)	FASE 2 (Squad Leader tenta entender o problema)	FASE 3 (Neymar tenta solucionar o problema)	FASE 4 (Análise do colaborador)	FASE 5 (Desfecho)
1 - Kaique comenta com um amigo de seu squad sobre sua insatisfação. 2 - O amigo decide contar para o Squad Leader que seu colega está considerando sair da empresa, como uma forma de impedir que isso aconteça.	1. O Squad Leader conversa com Neymar para entender suas dores. 2. Kaique não se sente ouvido pelo Squad Leader.	1. Kaique vai até o RH para reclamar de seu chefe, pois ele não foi receptivo. 2. O time de RH disse que poderia conversar com o líder do squad sobre o ocorrido.	1. O Squad Leader, juntamente com o time de RH, analisaram o desempenho de Neymar e ainda confirmaram ele no modelo. 2. O modelo mostrava que ele estava "propenso a sair". 3. Com as análises, o Squad Leader concluiu que o Kaique não merecia o reconhecimento que havia pedido.	1. Kaique não ficou satisfeito com a resposta que recebeu do seu Squad Leader. 2. O funcionário então, aceita outra proposta de emprego.
<b>Oportunidades</b>		<b>Responsabilidades</b>		
Considerando que o mercado está aquecido, muitos funcionários acabam super valorizando o seu trabalho, e as vezes exigem um maior reconhecimento do que apresentam nas entregas.		Essa situação mostra como o mercado de Dev's está aquecido e que se deve tomar cuidado nas contratações para conseguir reter a maior quantidade de funcionários possível.		

Figura 9. Análise e descrição do cenário e jornada do Kaique Roman

## 4.2. Compreensão dos Dados

### 4.2.1 Descrição dos dados

Nesta seção, destrinchamos os dados fornecidos dos arquivos de dados “Base Colaboradores Everymind\_Inteli\_2020 a 2022vModelo Preditivo” e “Novas Informações\_Eveymind\_27.09.22” fornecidos 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 alguns dos 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.

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.

#### → Variáveis

E  
I  
m  
e  
n  
t  
o  
s  
  
↓

	Matrícula	Nome Completo	Data Admissão	Data Saída	Tipo Saída	Cargo	Salário Mês	Data Nascimento	Gênero	Etnia	Estado Civil	Escolaridade	Estado	Cidade	Área
2	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
...	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
...	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
475	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
476	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Figura 10. Descrição da tabela Everymind

- Matrícula: A coluna contém o número de identificação do funcionário, definido por um número natural.
- Nome Completo
- Data de admissão: Data de entrada do colaborador na empresa, no formato “MM/DD/AAAA”.
- Data de saída (caso o funcionário não esteja mais na empresa): Data de saída do colaborador, no formato “MM/DD/AAAA”.

- Tipo de saída (caso o funcionário não esteja mais na empresa): Dispensa sem Justa Causa , Pedido de Demissão ou Rescisão Contrato Exp - Pedido.

- Cargo
- Salário mês: Último salário pago ao colaborador
- Data de nascimento: Data de nascimento do colaborador, no formato “MM/DD/AAAA”.
- Gênero: Masculino ou Feminino
- Etnia: Maioria não está informado
- Estado civil
- Escolaridade: Grau de escolaridade
- Área: Área de atuação na empresa
- Estado
- Cidade
- Idade

## 2. 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.

### → Variaveis

E l e m e n t o s	Matricula	Codinome	Situação	Data de Admissão	Data Vigência	Novo Cargo	Novo Salario	Motivo	Alterou Função
2	X	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X	X
..	..	..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..	..	..
339	X	X	X	X	X	X	X	X	X
340	X	X	X	X	X	X	X	X	X

Figura 11. Descrição da tabela Reconhecimento

- 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”.

### 3. 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.

#### E → Variaveis

	Divisao	Pilar	Pontuação	Fator	Pontuação	Pergunta	Porcentagem das respostas	Taxa de Confiabilidade
2	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
1695	X	X	X	X	X	X	X	X

Figura 12. Descrição da tabela “Ambiente de trabalho”

- 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. Performance

Nesta seção, descrevemos os dados fornecidos da sessão “Performance 20 a 22” 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 das avaliações feitas e avaliações do gestor recebidas de cada um dos colaboradores.

#### E → Variáveis

	Codinome	Auto Avaliacao 20	Avaliacao Gestor 20	Auto Avaliacao 21	Avaliacao Gestor 21	Auto Avaliacao 22	Avaliacao Gestor 22
m	X	X	X	X	X	X	X
e	X	X	X	X	X	X	X
n	...	...	...	...	...	...	...
t	X	X	X	X	X	X	X
o	X	X	X	X	X	X	X
s	X	X	X	X	X	X	X

Figura 13. Descrição da tabela “Performance”

- ↓
- Codinome - Referente ao número do colaborador em questão definido por um número natural.
  - Auto Avaliação 20 - Auto Avaliação de cada colaborador referente ao ano de 2020, é definida por um número natural em um intervalo de 0 a 5.
  - Avaliação Gestor 20 - Avaliação do gestor para cada colaborador referente ao ano de 2020, é definida por um número natural em um intervalo de 0 a 5.
  - Auto Avaliação 21 - Auto Avaliação de cada colaborador referente ao ano de 2021, é definida por um número natural em um intervalo de 0 a 5.
  - Avaliação Gestor 21 - Avaliação do gestor para cada colaborador referente ao ano de 2021, é definida por um número natural em um intervalo de 0 a 5.
  - Auto Avaliação 22 - Auto Avaliação de cada colaborador referente ao ano de 2022, é definida por um número natural em um intervalo de 0 a 5.
  - Avaliação Gestor 22 - Avaliação do gestor para cada colaborador referente ao ano de 2022, é definida por um número natural em um intervalo de 0 a 5.

## 5. Horas Extras

Nesta seção, destrinchamos os dados fornecidos da sessão “Performance 20 a 22” 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 quantidade e valor das horas extras feitas por cada um dos colaboradores.

E → Variaveis

Codinome	DEZ19 A JUL20 - Qtde	DEZ19 A JUL20 - Valor	...	ABR A JUL22 - Qtde	ABR A JUL22 - Valor
X	X	X	...	X	X
X	X	X	...	X	X
...	...	...	...	...	...
X	X	X	...	X	X
X	X	X	...	X	X

Figura 14. Descrição da tabela “Horas Extras”

- Codinome - Referente ao número do colaborador em questão definido por um número natural.
- DEZ19 A JUL20 - Qtde - Referente a quantidade de horas extras feitas no período de dezembro de 2019 a julho de 2020, definido por um número inteiro.
- DEZ19 A JUL20 - Valor - Referente aos valores das horas extras do período de dezembro de 2019 a julho de 2020, definido por um número inteiro.
- AGO A NOV20 - Qtde - Referente a quantidade de horas extras feitas no período de agosto a novembro de 2020, definido por um número inteiro.
- AGO A NOV20 - Valor - Referente aos valores das horas extras do período de agosto a novembro de 2020, definido por um número inteiro.
- DEZ20 A MAR21 - Qtde - Referente a quantidade de horas extras feitas no período de dezembro de 2020 a março de 2021, definido por um número inteiro.
- DEZ20 A MAR21 - Valor - Referente aos valores das horas extras do período de dezembro de 2020 a março de 2021, definido por um número inteiro.
- ABR A JUL21 - Qtde - Referente a quantidade de horas extras feitas no período de abril a julho de 2021, definido por um número inteiro.
- ABR A JUL21 - Valor - Referente aos valores das horas extras do período de abril a julho de 2021, definido por um número inteiro.
- AGO A NOV21 - Qtde - Referente a quantidade de horas extras feitas no período de agosto a novembro de 2021, definido por um número inteiro.
- AGO A NOV21 - Valor - Referente aos valores das horas extras do período de agosto a novembro de 2021, definido por um número inteiro.
- DEZ21 A MAR22 - Qtde - Referente a quantidade de horas extras feitas no período de dezembro de 2021 a março de 2022, definido por um número inteiro.
- DEZ21 A MAR22 - Valor - Referente aos valores das horas extras do período de dezembro de 2021 a março de 2022, definido por um número inteiro.

- ABR A JUL22 - Qtde - Referente a quantidade de horas extras feitas no período de abril a julho de 2022, definido por um número inteiro.
- ABR A JUL22 - Valor - Referente aos valores das horas extras do período de abril a julho de 2022, definido por um número inteiro.

#### **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. Além disso, os dados trabalhados são locais e foram cedidos pela empresa, respeitando o LGPD.

## 4.2.2 Descrição estatística básica dos dados

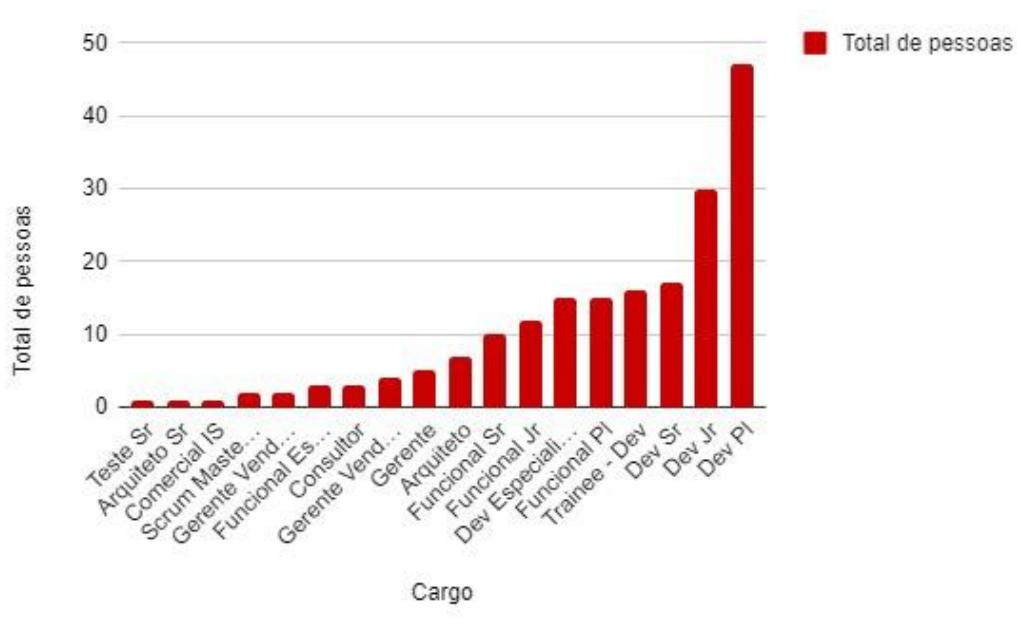


Gráfico 1 - Total de pessoas que saíram de acordo com o cargo que exerciam

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.

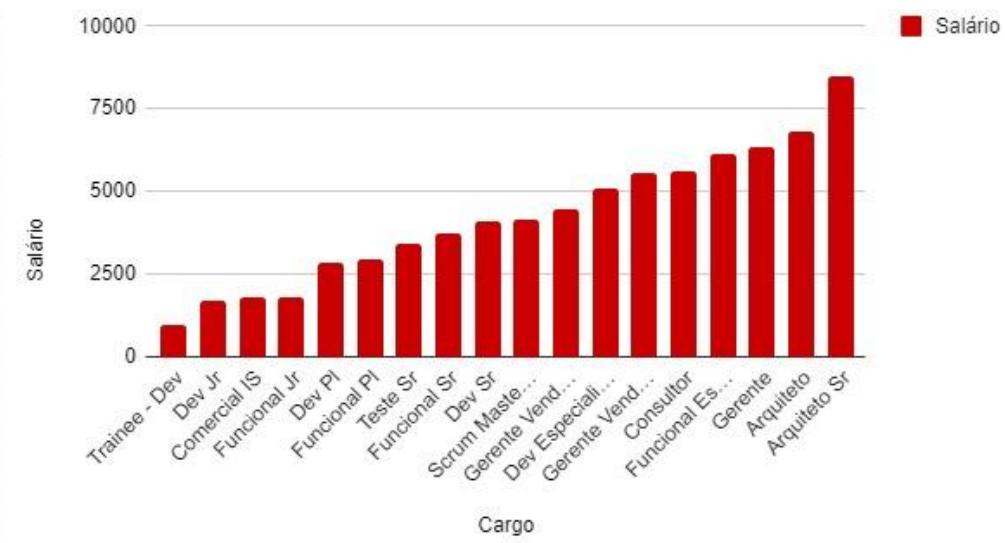


Gráfico 2 - Média salarial das pessoas que saíram pelo cargo que exerciam

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.

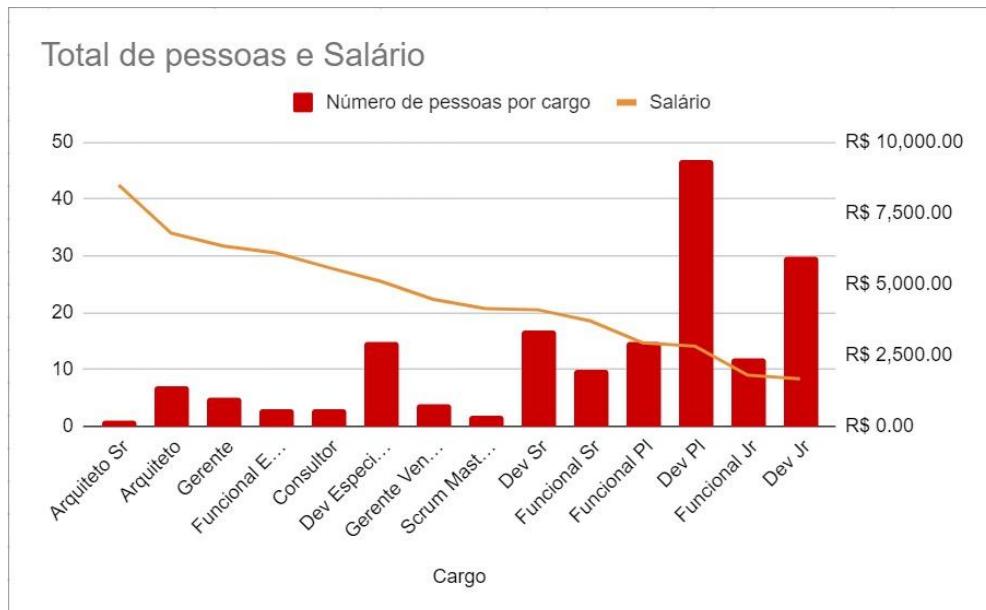


Gráfico 3 - Total de pessoas que saíram da empresa com relação ao salário que ganhavam

Esse gráfico é uma junção do gráfico 1 e 2, e facilita a visualização da diferença entre a quantidade de pessoas que saíram da empresa e o salário que elas ganhavam.

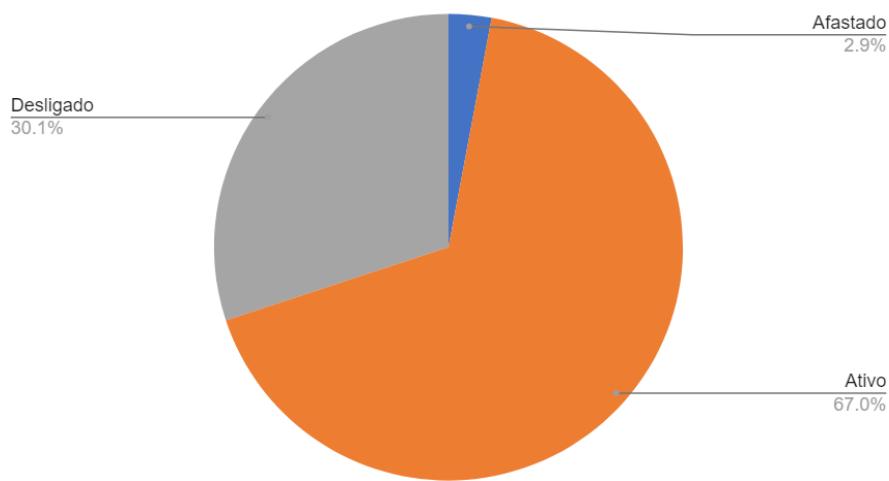


Gráfico 4 - Reações pós promoção: Colaboradores desligados VS Afastados VS Ativos

A hipótese que pode ser extraída desse gráfico e que se relaciona com o gráfico 2, é que, considerando que  $\frac{1}{3}$  da empresa saiu mesmo após uma ou mais promoções, os funcionários continuam recebendo ofertas de trabalho que consideraram ter uma melhor remuneração e, por isso, consideraram sair da empresa mesmo após ter tido algum tipo de reconhecimento.

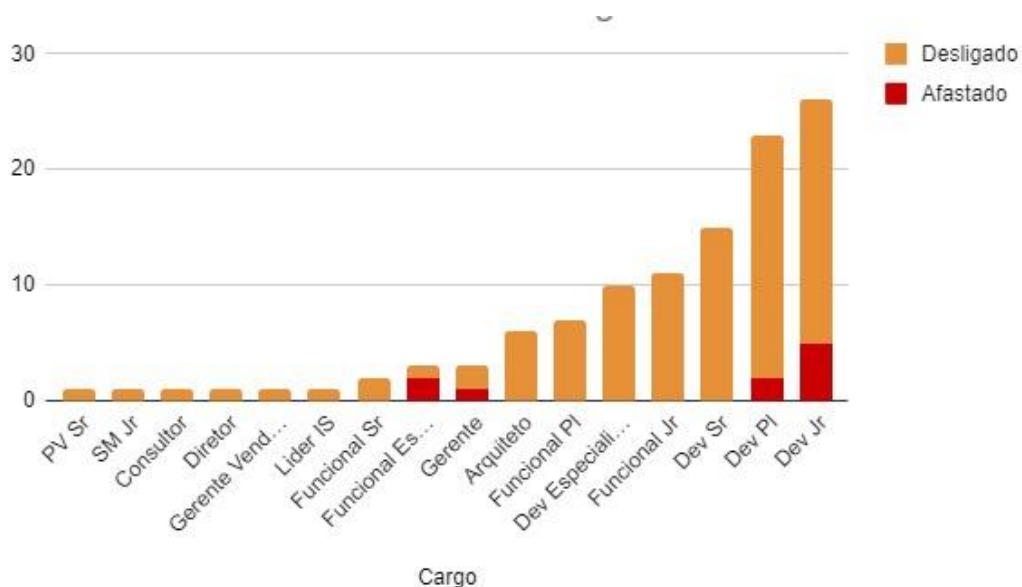
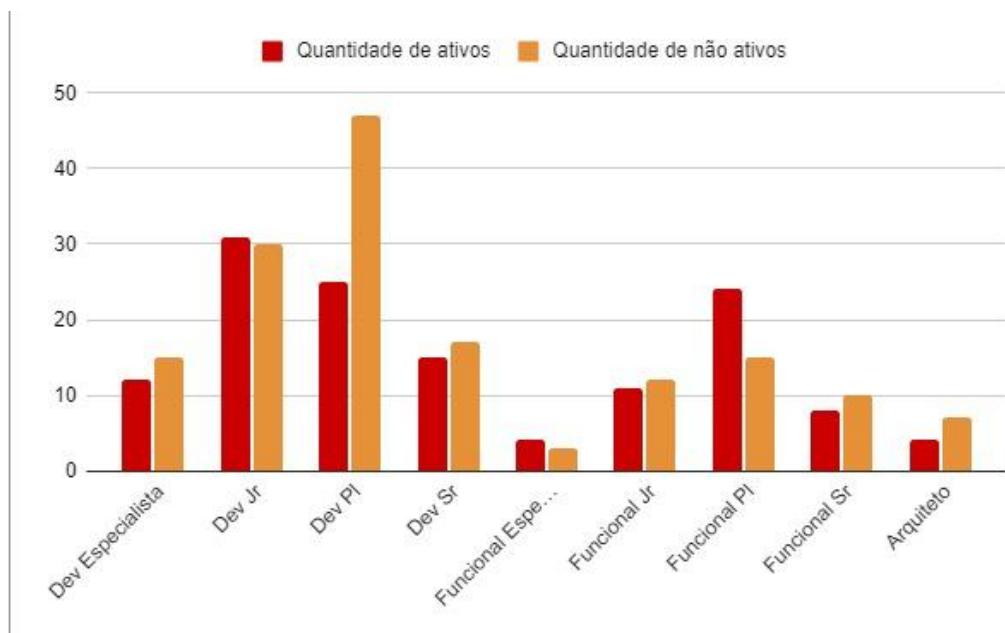
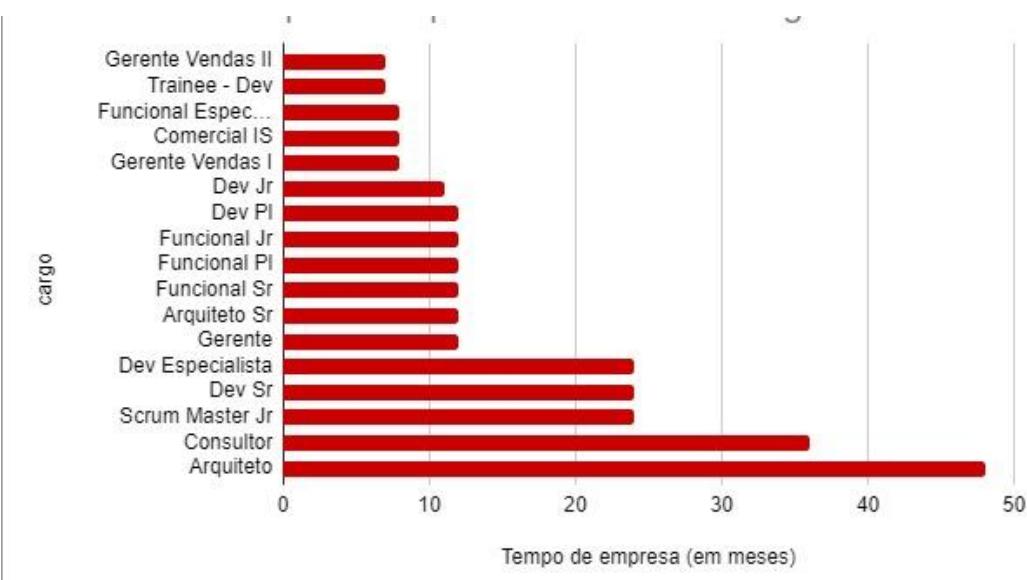


Gráfico 5 - Demissões pós promoção em relação ao cargo exercido

O gráfico acima mostra que, das pessoas que receberam promoções e saíram (gráfico 4), a maioria são desenvolvedores, o que reforça a nossa hipótese de que os Devs costumam rotacionar no mercado com mais frequência.



Analizando esse gráfico, é possível perceber que a quantidade de Devs que saíram da empresa supera os que ainda trabalham nela atualmente (com exceção dos Devs Jr), o que é alarmante, e reforça a ideia de que é preciso se preocupar com o Turnover desses colaboradores, uma vez que, de acordo com as análises dos gráficos anteriores, eles recebem muitas ofertas e trabalho, o que torna ainda mais difícil para a empresa contratar funcionários que exerçam esse cargo.



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 categorias já estão pré-definidas (“Saiu” ou “não saiu”) e o algoritmo vai definir qual colaborador se encaixa em cada classe, ou seja, a predição desejada é saber se o colaborador está propenso ou não a sair da empresa. 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. As tabelas a seguir visam replicar os resultados obtidos das análises preliminares.

**Tabela “Everymind”**

#	Coluna	Contador de valores não-nulos	Tipo do dado
0	Matrícula	475 non-null	float64
1	Nome Completo	475 non-null	object
2	Dt Admissao	475 non-null	object
3	Dt Saida	191 non-null	datetime64 [ns]
4	Tipo Saida	191 non-null	object
5	Cargo	475 non-null	object
6	Salario Mês	475 non-null	float64
7	Dt Nascimento	475 non-null	datetime64 [ns]

8	Genero	475 non-null	object
9	Etnia	475 non-null	object
10	Estado Civil	475 non-null	object
11	Escolaridade	475 non-null	object
12	Estado	475 non-null	object
13	Cidade	475 non-null	object
14	Area	475 non-null	object

**Tabela “Reconhecimento”**

#	Coluna	Contador de valores não-nulos	Tipo do dado
0	Matricula	339 non-null	float64
1	Codinome	339 non-null	object
2	Situação	339 non-null	object
3	Data de Admissão	339 non-null	datetime64[ns]
4	Data Vigência	339 non-null	datetime64[ns]
5	Novo Cargo	339 non-null	object
6	Novo Salario	339 non-null	object
7	Motivo	339 non-null	object
8	Alterou Função	339 non-null	object(5)

**Tabela “Ambiente de Trabalho”**

#	Coluna	Contador de valores não-nulos	Tipo do dado
0	Divisao	1694 non-null	object
1	Pilar	1694 non-null	object
2	Pontuação	1694 non-null	object

3	Fator	1694 non-null	object
4	Pontuação.1	1694 non-null	object
5	Pergunta	1694 non-null	object
6	Pulou	243 non-null	object
7	Muito Insatisfeito	126 non-null	object
8	Insatisfeito	289 non-null	object
9	Neutro	422 non-null	object
10	Satisffeito	1018 non-null	object
11	Muito Satisffeito	1204 non-null	object
12	Taxa de Confiabilidade	1694 non-null	object

**Tabela “Performance”**

#	Coluna	Contador de valores não-nulos	Tipo do dado
0	Codinome	251 non-null	object
1	Auto Avaliacao 20	91 non-null	float64
2	Avaliacao Gestor 20	65 non-null	float64
3	Auto Avaliacao 21	104 non-null	float64
4	Avaliacao Gestor 21	102 non-null	float64
5	Auto Avaliacao 22	94 non-null	float64
6	Avaliacao Gestor 22	131 non-null	float64

**Tabela “Horas Extras”**

#	Colunas	Contador de valores não-nulos	Tipo do dado
0	Codinome	475 non-null	object
1	DEZ19 A JUL20 -	73 non-null	float64

	Qtde		
2	DEZ19 A JUL20 - Valor	73 non-null	float64
3	AGO A NOV20 - Qtde	55 non-null	float64
4	AGO A NOV20 - Valor	55 non-null	float64
5	DEZ20 A MAR21 - Qtde	56 non-null	float64
6	DEZ20 A MAR21 - Valor	56 non-null	float64
7	ABR A JUL21 - Qtde	75 non-null	float64
8	ABR A JUL21 - Valor	75 non-null	float64
9	AGO A NOV21 - Qtde	78 non-null	float64
10	AGO A NOV21 - Valor	78 non-null	float64
11	DEZ21 A MAR22 - Qtde	132 non-null	float64
12	DEZ21 A MAR22 - Valor	132 non-null	float64
13	ABR A JUL22 - Qtde	102 non-null	float64
14	ABR A JUL22 - Valor	102 non-null	float64

#### 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.

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.

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 nas abas de “Reconhecimento”, “Performance 20 a 22” e “HEs\_2020 a 22” na coluna “Codinome”. Essa Feature foi selecionada para padronizar todos os dados categóricos em números e facilitar análises futuras do algoritmo.

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.

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 .

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.

#### **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 Saída” e “Tipo Saída” 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. A decisão de remover ou substituir registros em branco 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”.

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).

Da aba “Performance 20 a 22” os valores vazios significam que o colaborador da linha em questão não havia feito uma autoavaliação e recebido uma avaliação do gestor, então foi atribuído o valor 0 às colunas vazias.

Da aba “HEs\_2020 a 22” os valores vazios significam que o colaborador da linha em questão não fez nenhuma hora extra e portanto não recebeu um valor referente às horas, então foi atribuído o valor 0 às colunas vazias.

#### **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.

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 atualmente só informa se o colaborador é ou não desenvolvedor, foi feito também uma coluna de média salarial

que diz se o colaborador está ou não acima da média salarial da empresa, uma outra que aponta a quantidade de horas extras feitas por cada colaborador e o valor delas 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 em questão de acuracidade e precisão.

Foi identificada a necessidade de criar colunas referentes a avaliações do colaborador e da área em que ele trabalha, foram gerados os seguintes dados: avaliação dos gestores e número total de avaliações, autoavaliação e número total de autoavaliações e a nota da área baseado na pesquisa de ambiente de trabalho.

#### 4.3.5 Colunas não utilizadas

Na etapa de análise e escolha dos dados, foi excluída a coluna referente à etnia dos colaboradores. A maioria das células encontravam-se vazias, o que dificultaria a análise imparcial dos dados e acarretaria em extrações de informações enviesadas. Dada a importância da aplicação da ética em sistemas de informação, optamos por não utilizar estes dados a fim de resguardar a integridade moral dos colaboradores e garantir uma solução pouco viciada (overfitting) aos dados de treino.

#### 4.3.6 Mesclar conjuntos de dados e registros

Ao fim da formatação, categorização e padronização do banco de dados, todas as informações foram transferidas 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.

#	Coluna	Contador de valores não-nulos	Tipo do dado
0	NumeroMeses	475 non-null	int64
1	Idade	475 non-null	int64
2	Salario Mês	475 non-null	float64
3	Cargo	475 non-null	uint8
4	Genero	475 non-null	uint8
5	Estado Civil	475 non-null	uint8

6	Area	475 non-null	uint8
7	Remoto	475 non-null	int64
8	mediaTempoPromoção	475 non-null	float64
9	dev	475 non-null	int64
10	AutoAvaEverymind	475 non-null	float64
11	numAutoAvaEv	475 non-null	int64
12	GestoAvaEverymind	475 non-null	float64
13	numGestoAvaEv	475 non-null	int64
14	qtndHorasEver	475 non-null	float64
15	ValorHorasEver	475 non-null	float64
16	notaArea	475 non-null	float64

Obs: Colunas do tipo “uint8” é correspondente a números inteiros.

#### 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 Naïve Bayes, que estava com 63% e aumentou para 70%, o Support Vector Machine 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, pois o atributo 'Salário Mês' possui valores discrepantes se comparado aos outros atributos.

### 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, a classificação de dados será determinada de acordo com a distância dos "vizinhos" mais próximos. Também utiliza o método 'lazy learning', por isso não é necessário treinar o modelo, pois a ideia desse método é tratar com bases de dados que serão constantemente atualizadas.

Assumindo que coisas semelhantes estão próximas umas das outras, o algoritmo calcula a distância entre N dimensões, podendo utilizar diversas técnicas: Euclidiana, Hamming, Manhattan, Mahalanobis e Minkowski.

A técnica utilizada foi a Distância Euclidiana, 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:

$$E(x, y) = \sqrt{\sum_{i=0}^n (x_i - y_i)^2}$$

Figura 15. Visualização da fórmula do K-Nearest-Neighbor

#### Distância Euclidiana

$$d_{AB} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

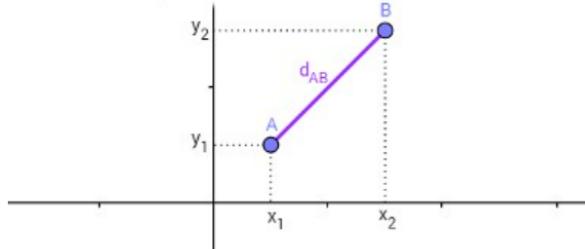


Figura 16. Visualização da fórmula da distância Euclidiana

### 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.

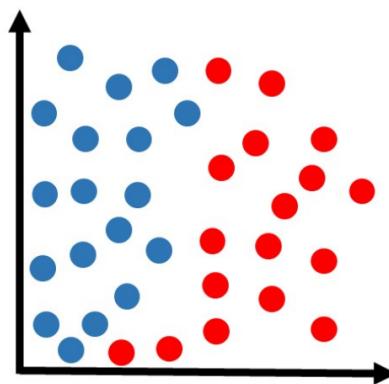


Figura 17. Exemplificação do K-Nearest-Neighbor

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 Naïve Bayes

O algoritmo Naïve 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, porém a suposição de independência entre as variáveis nem sempre se confirma em problemas reais.

Este algoritmo funciona bem com variáveis categóricas e para o caso das variáveis numéricas, ele as considera como distribuições normais. Por outro lado, se uma variável categórica tem uma classe contida no conjunto de teste mas não observada no conjunto de treino, um valor nulo de probabilidade lhe será atribuído, o que impossibilita uma predição e técnicas de suavização deverão ser aplicadas.

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

Figura 18. Fórmula do algoritmo de Naïve Bayes

### 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. Este é um algoritmo de aprendizado de máquina supervisionado para classificação e para regressão

A árvore 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.

O cálculo que define a entropia é dado por:

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i$$

Figura 19. Fórmula da entropia da Árvore de decisão

Em que  $p_i$  é a probabilidade de um elemento ser pertencente a uma dada classe.

Na ligação dos nós é aplicado 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 a partir da árvore gerada 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.

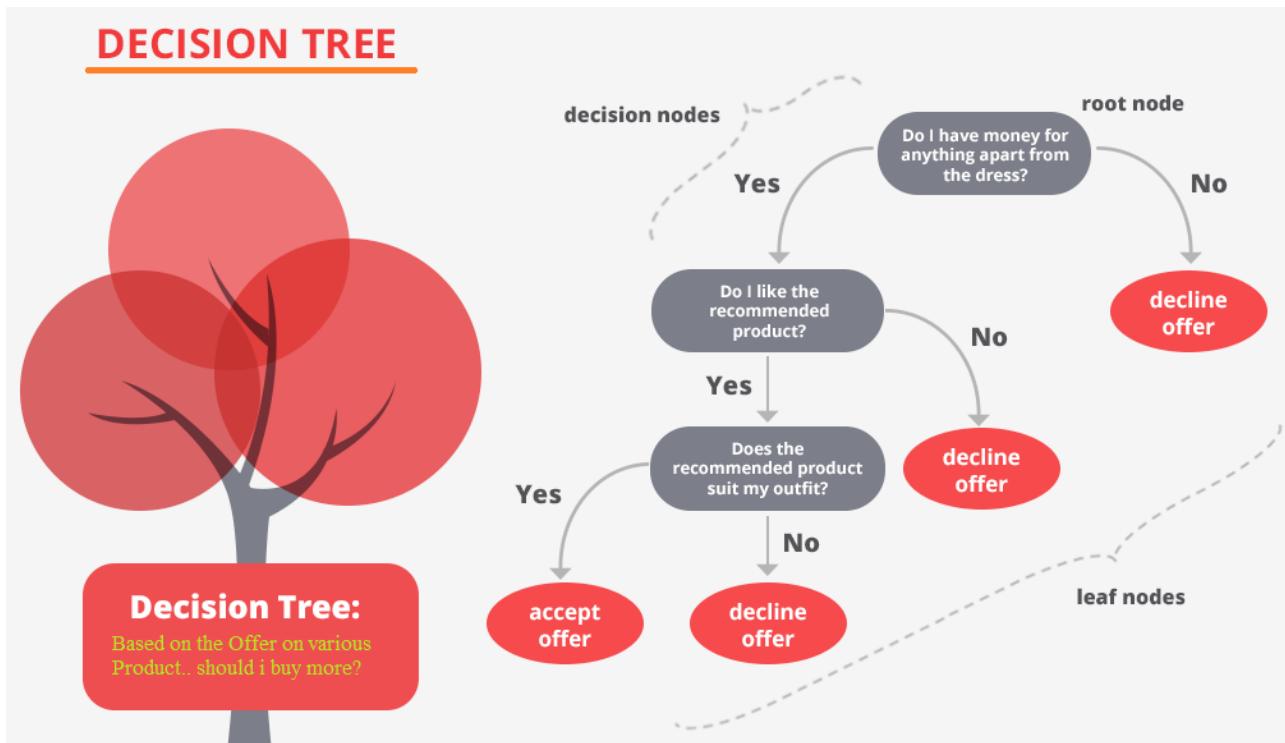


Figura 20: Exemplificação da árvore de decisão

#### 4.4.4 Random Forest

O Random Forest é um algoritmo preditivo que constrói um grande número de árvores de decisão, realizando escolhas aleatórias entre determinadas variáveis para definir cada nó criado.

O modelo possui três hiperparâmetros principais que precisam ser definidos antes do treinamento da amostra: tamanho do nó, número de árvores e número de recursos. Após essas definições, o modelo cria e compara as árvores entre si, selecionando a que tiver uma melhor avaliação.

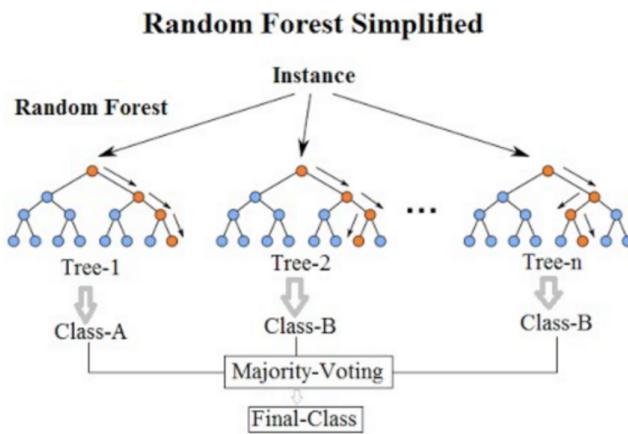


Figura 21. Exemplificação do Random Forest

## 4.4.5 Support Vector Machine

Modelo Support Vector Machine: O Support Vector Machines (SVM - Máquinas de Vetores de Suporte, em tradução literal) 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.

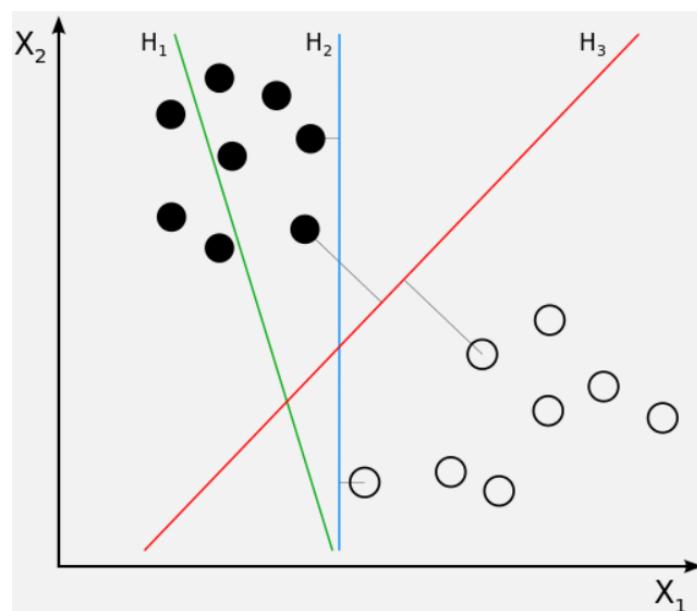


Figura 22. Exemplificação do Support Vector Machine

Fonte: Modelos de Predição I 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  $x_i \in \mathbb{R}^p$ ,  $i=1,\dots,n$ , em duas classes e um vetor  $y \in \{-1,1\}^n$ , a meta é descobrir  $w \in \mathbb{R}^p$  e  $b \in \mathbb{R}$  tal que a previsão dada por  $\text{sign}(w^T \Phi(x) + b)$  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:

$$\begin{aligned} & \min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^n \zeta_i \\ & \text{subject to } y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i, \\ & \quad \zeta_i \geq 0, i = 1, \dots, n \end{aligned}$$

Figura 23. Fórmula da resolução do Support Vector Machine

Tentamos então maximizar as margens enquanto aplicamos pequenas punições à amostras classificadas incorretamente ou próximas à margem. Há aceitação de uma certa distância  $\zeta_i$  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.

## 4.4.6 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.

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

O gráfico dessa função tem o seguinte formato:

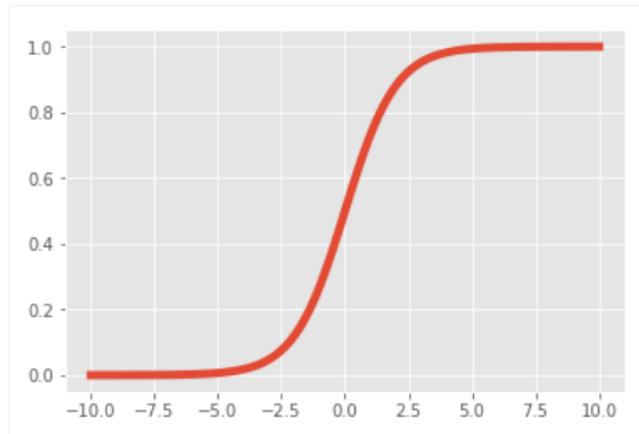


Figura 24. Exemplificação e fórmula para a regressão logística

## 4.4.7 Justificativa das escolhas dos algoritmos

Escolhemos esses algoritmos devido a alta usabilidade no mercado de trabalho e também pelo fato de ser facilmente aplicável. No final escolhemos os seguintes algoritmos: K-Nearest-Neighbor, Naïve Bayes, Árvore de Decisão, Support Vector Machine, Random Forest e Regressão Logística. A melhor acurácia dentre os citados foi encontrada no algoritmo de Regressão Logística.

## 4.5. Avaliação

### 4.5.1. Features utilizadas

Para treinar todos os modelos, foi utilizado um conjunto de colunas que foram julgadas importantes, sendo elas: “NumeroMeses”, “Idade”, “Salario Mês”, “Cargo”, “Genero”, “Estado Civil”, “Area”, além das criadas depois devido a necessidade (derivação de novos atributos): “Remoto”, “mediaTempoPromoção”, “dev”, “AutoAvaEverymind”, “numAutoAvaEv”, “GestoAvaEverymind”, “numGestoAvaEv”, “qntdHorasEver”, “ValorHorasEver” e “notaArea”. A princípio, uma matriz de correlação foi usada para auxiliar na escolha dos atributos. Após a aplicação dos modelos, geramos um gráfico usando “valores de Shapley” para avaliar a importância de cada uma dessas variáveis. Os resultados são apresentados a seguir:

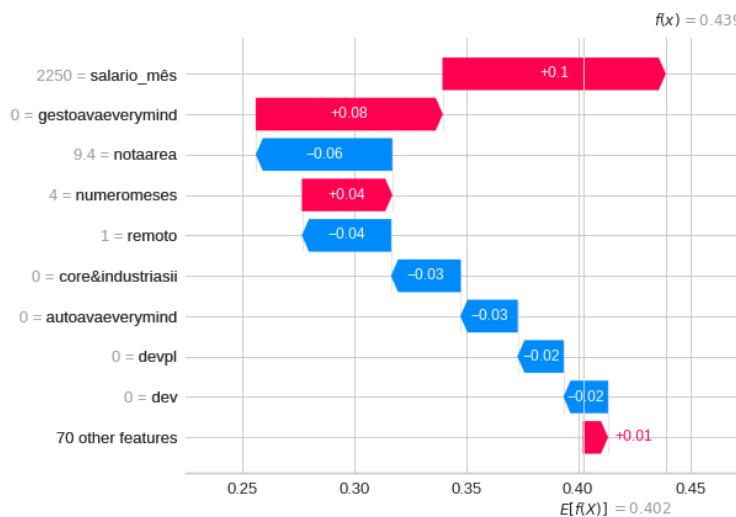


Figura 24. Descrição das features de mais impacto no modelo

Em que as features contêm pontuações que indicam a contribuição nos resultados classificatórios. Nesse caso, as setas, indicadas pelas cores vermelha e azul, indicam se a variável é mais impactante para uma ou outra classificação (“Saiu”/“Não Saiu”).

#### 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. Validação cruzada

Validação cruzada é um processo que utiliza diferentes amostras para treino e teste, a fim de melhorar a predição. É possível escolher em quantas partes o modelo irá dividir os dados. Esse processo foi aplicado em nosso modelo com o objetivo de obter uma média de acurácia de cada um para várias amostras diferentes e, considerando que dividimos os dados em cinco partes, percebemos que a média dos modelos varia entre 51% e 72%, sendo que o modelo com menor média é o Naïve Bayes, e o com maior é o SVM.

Apesar disso, por mais que a média de alguns modelos tenham sido baixas, eles performam melhor individualmente, como a árvore de decisão.

Modelo de predição	Média dos valores resultantes da validação cruzada
Naïve Bayes	58%
Knn	68%
Random forest	75%

Regressão logística	72%
Árvore de decisão	72%
SVM	79%

#### 4.5.4. Métricas de avaliação

Para definir o desempenho de um algoritmo e, consequentemente, seu sucesso ou fracasso, utilizamos métricas estatísticas que auxiliam a avaliar os resultados gerados. O sucesso de um algoritmo/método/classificador é definido pela análise dos resultados do conjunto de métricas que são definidas abaixo, principalmente pela pontuação de sua acurácia, da área abaixo da curva ROC e dos resultados da matriz de confusão. Somado a isso, a escolha para a utilização dessas métricas permite a comparação dos resultados gerados pelos diferentes modelos.

##### 4.5.4.1. 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.

$$\text{acurácia} = \frac{VP + VN}{VP + FN + VN + FP}$$

Figura 25. Fórmula da acurácia

##### 4.5.4.2. Recall

Usando o cálculo de recall sabemos o quanto 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.

$$recall = \frac{TP}{TP+FN}$$

Figura 26. Fórmula do Recall

##### 4.5.4.3. Precisão

Mostra qual proporção de identificações positivas foi realmente correta.

$$precisão = \frac{VP}{VP + FP}$$

Figura 27. Fórmula da precisão

#### 4.5.4.4. 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.

#### 4.5.4.5. 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.

#### 4.5.4.6. 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.

#### 4.5.4.7. Curva ROC

A curva ROC (*Receiver Operating Characteristic ou, em tradução livre, Característica de Operação do Receptor*) é um método utilizado para avaliar e validar os resultados dos modelos testados. Utilizando um plano cartesiano, definimos o eixo X como Sensibilidade, que descreve capacidade de detectar verdadeiros eventos na amostra e é dada por:  $Nº\ de\ Verdadeiros\ Negativos/(Nº\ de\ Verdadeiros\ Negativos + Nº\ de\ Falsos\ Positivos)$  e o eixo Y como Especificidade, que descreve a capacidade de detectar algum evento na amostra e é dada por:  $Nº\ de\ Verdadeiro\ Positivo/(Nº\ de\ Verdadeiro\ Positivo + Nº\ de\ Falso\ Negativo)$ . A partir dos valores de Especificidade e Sensibilidade calculados para vários modelos, geramos pontos no gráfico que descrevem a relação entre os dois valores para um dado teste e, a partir deles, conseguimos desenhar uma curva que evidencia o desempenho geral de um método. Quanto maior a área sob a curva, melhor sua performance, pois terá uma grande Especificidade e uma pequena Sensibilidade (valores mais à esquerda e acima).

$$TPR = \frac{TP}{TP + FN} \qquad FPR = \frac{FP}{TN + FP}$$

Figura 28. Fórmula para resultado de falsos negativos e verdadeiros positivos

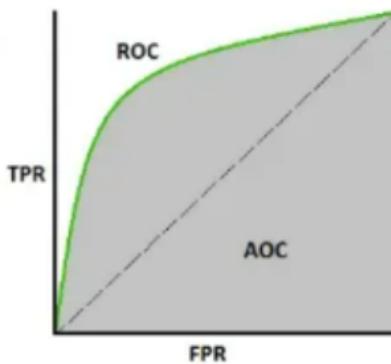


Figura 29. Exemplificação da Curva ROC

#### 4.5.4.8. Matriz de confusão

A matriz de confusão é uma tabela que relaciona as previsões do modelo com os dados reais da tabela analisada, dividindo os valores entre falsos positivos, falsos negativos, verdadeiros positivos e verdadeiros negativos. Assim, ela indica qual a qualidade do modelo previsor atual.

Na prática, os valores previstos são mostrados no eixo X e no eixo Y os valores reais da base de dados. Assim, os acertos ficam onde o eixo X e o eixo Y apresentam o mesmo valor.

		Valor Preditivo	
		Sim	Não
Real	Sim	Verdadeiro Positivo (TP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (TN)

Figura 30. Exemplificação dos valores verdadeiros positivos e falsos negativos

#### 4.5.4.9. Normalização, Padronização e oversampling

Durante a execução dos modelos, testamos a aplicação de métodos para padronização e normalização para verificar se havia melhora nos resultados. Após diversas análises de performance, decidimos mantê-los os métodos nos quais houve aumento de performance, tanto de precisão como no tempo de execução. Para isso, mantivemos os oversampling nos classificadores “Naïve Bayes”, “Random Forest”, “Support Vector Machine”. Da mesma forma, decidimos padronizar os dados nos modelos “Random Forest”, “Support Vector Machines” e “Regressão Logística”. Normalizamos os dados para os modelos “K-Nearest Neighbors”, “Árvore de Decisão”, “Random Forest”.

## 4.5.5. Resultado das métricas de avaliação

### 4.5.5.1 K Nearest Neighbor

#### 4.5.5.1.1 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): 76% e de Acurácia (teste): 65%.

	Precision	Recall	f1-score	support
0	0.68	0.75	0.71	81
1	0.62	0.53	0.57	62
Accuracy			0.66	143
Macro avg	0.65	0.64	0.64	143
Weighted avg	0.65	0.66	0.65	143

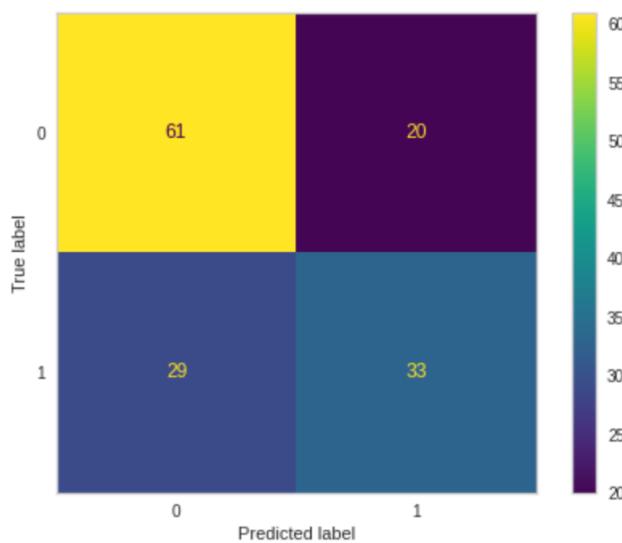


Figura 31. Matriz de confusão inicial para o K Nearest Neighbor

A partir da matriz de confusão do modelo K Nearest Neighbor, foi observado que a acuracidade foi mediana, uma vez que dos 62 funcionários que saíram da empresa, o modelo acertou 33 e dos 81 que não saíram da empresa o modelo acertou 61.

#### 4.5.5.1.2 Modelo com hiperparâmetros

Após o teste do modelo K Nearest Neighbor 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 K Nearest Neighbor, 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: 64%. Como é possível observar, a acurácia diminuiu 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']

#### Resultados obtidos:

	Precision	Recall	f1-score	support
0	0.64	0.80	0.71	81
1	0.62	0.42	0.50	62
Accuracy			0.64	143
Macro avg	0.63	0.61	0.61	143
Weighted avg	0.63	0.64	0.62	143

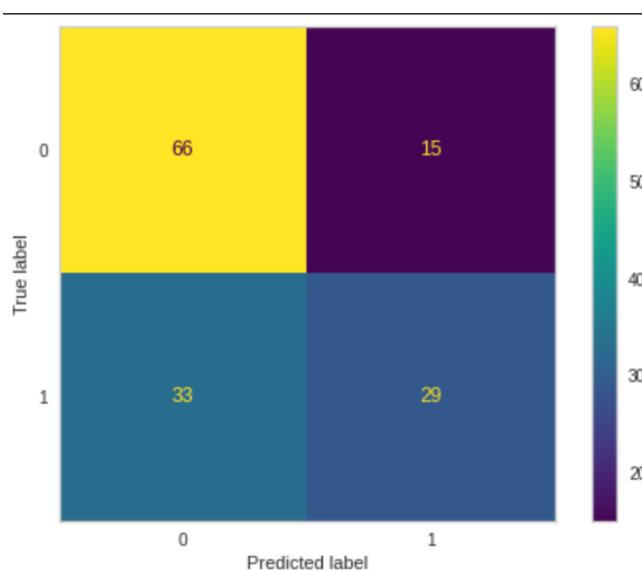


Figura 32. Matriz de confusão final para K Nearest Neighbor

Após a aplicação de hiperparâmetros, é possível perceber melhor que o modelo piorou, pois de 81 pessoas que não saíram, ele acertou 15, e de 62 colaboradores que saíram, o modelo acertou 29.

#### 4.5.5.1.3 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 (10, 20), podemos perceber qual a média de erro do nosso modelo, que no caso do K Nearest Neighbor é de 11%

### 4.5.5.2 Naïve Bayes

#### 4.5.5.2.1 Modelo default

Ao aplicar o Algoritmo Naïve Bayes, considerando as Features utilizadas, e os dados separados entre treino e teste, ele gerou um resultado de Acurácia (treino): 55% e de Acurácia (teste): 61%.

As imagens abaixo ajudam a compreender melhor os resultados:

	Precision	Recall	f1-score	support
0	0.73	0.30	0.42	81
1	0.59	0.90	0.71	90
Accuracy			0.61	171
Macro avg	0.66	0.60	0.57	171

Weighted avg	0.65	0.61	0.57	171

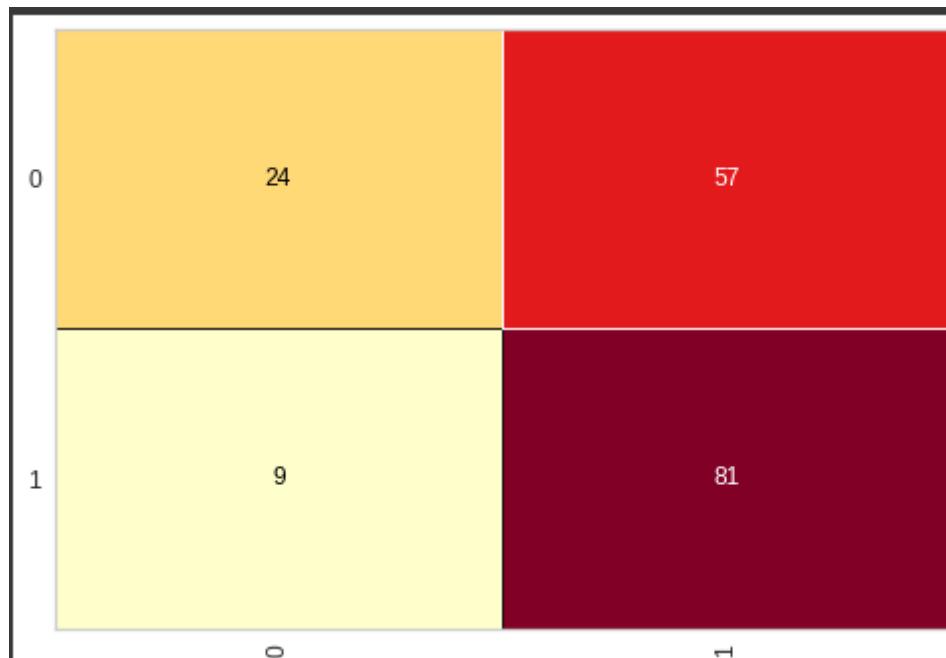


Figura 33. Matriz de confusão inicial para o Naïve Bayes

Observando a matriz de confusão do modelo de Bayes, pode-se concluir que o algoritmo obteve 24 acertos de 81 previsões das pessoas que continuam na empresa e 81 acertos de 90 previsões das pessoas que saíram da empresa, ou seja, 09 foram falsos negativos e 57 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

#### 4.5.5.2.2 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). No caso do Naïve Bayes, foi utilizado o RandomSearch, o melhor resultado que tivemos foi utilizando o seguinte hiperparâmetro: var\_smoothing: 0.000000007935, 0.000000008, 0.0000000085, 0.000000009, 0.000000009, 0.0000000009, e a acurácia obtida foi de: 67%. Como é possível observar, a acurácia diminuiu comparada ao uso de parâmetros default.

- Random Search

Parâmetros	Valor
var_smoothing	0.000000007935, 0.000000008, 0.0000000085, 0.000000009, 0.000000009, 0.0000000009

#### 4.5.5.2.3 Variância de erro do modelo

No treinamento do modelo, é pedido um parâmetro chamado "random state". O modelo foi testado com números em um intervalo de 0 a 9. 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 Bayes é de 6,3%. Segue um exemplo de 5 intervalos abaixo:

Acuracidade (teste): 5	0.71
Acuracidade (teste): 2	0.64
Acuracidade (teste): 4	0.65
Acuracidade (teste): 6	0.63
Acuracidade (teste): 8	0.64

De acordo com o intervalo apresentado anteriormente podemos concluir que a amostra que mais nos dá a maior classificação utiliza o valor "5", apresentando uma acuracidade de 71%.

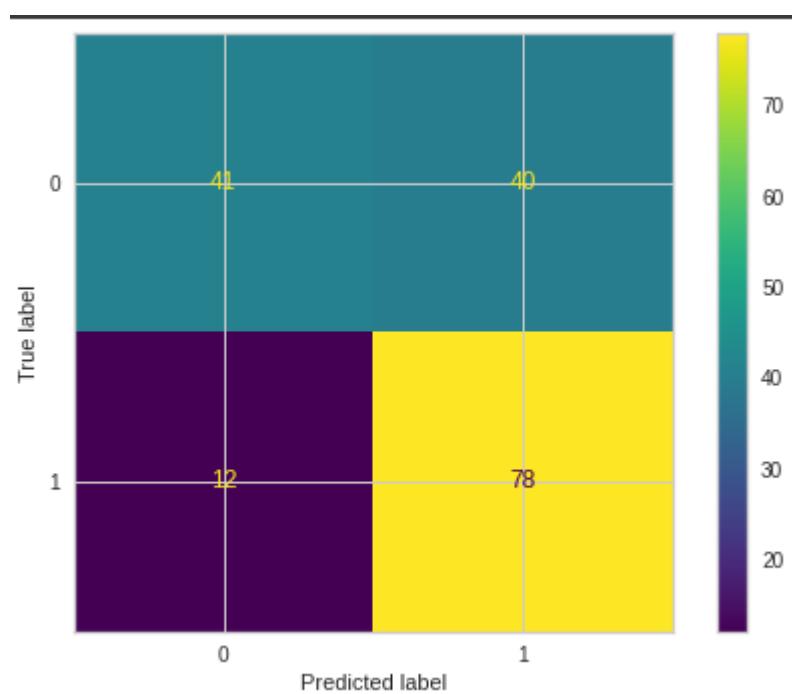


Figura 34. Matriz de confusão final para o Naïve Bayes

#### 4.5.5.3 Árvore de decisão

##### 4.5.5.3.1 Modelo default

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.

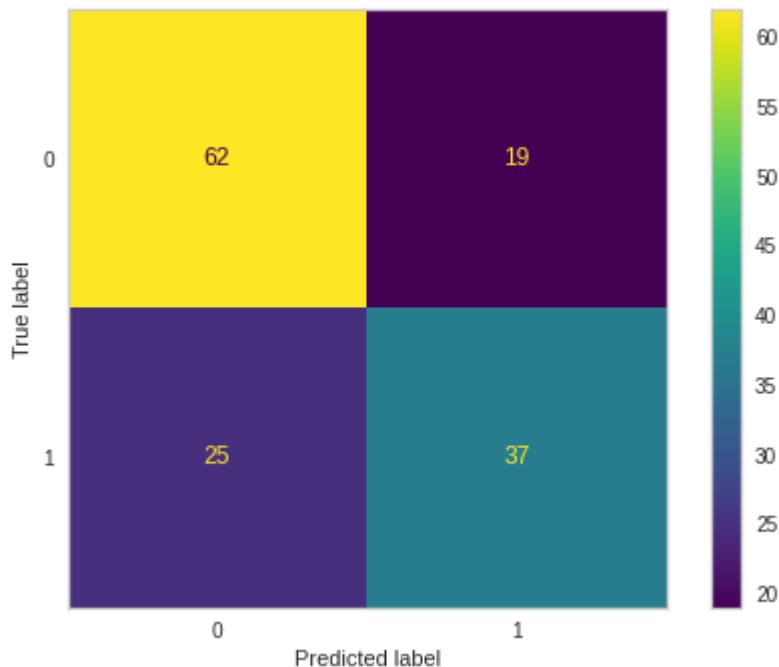


Figura 35: Matriz de confusão inicial para a Árvore de decisão

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.

#### 4.5.5.3.2 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

#### 4.5.5.3 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	0.6083916083916084
Acuracidade (teste): 71	0.6923076923076923
Acuracidade (teste): 72	0.5594405594405595
Acuracidade (teste): 73	0.5594405594405595
Acuracidade (teste): 74	0.5874125874125874

O intervalo apresentado anteriormente foi apenas uma amostra de todos os parâmetros que testamos, analisando todos podemos concluir que a amostra que mais nos dá a maior classificação é o de 90 com uma acuracidade de 73%.

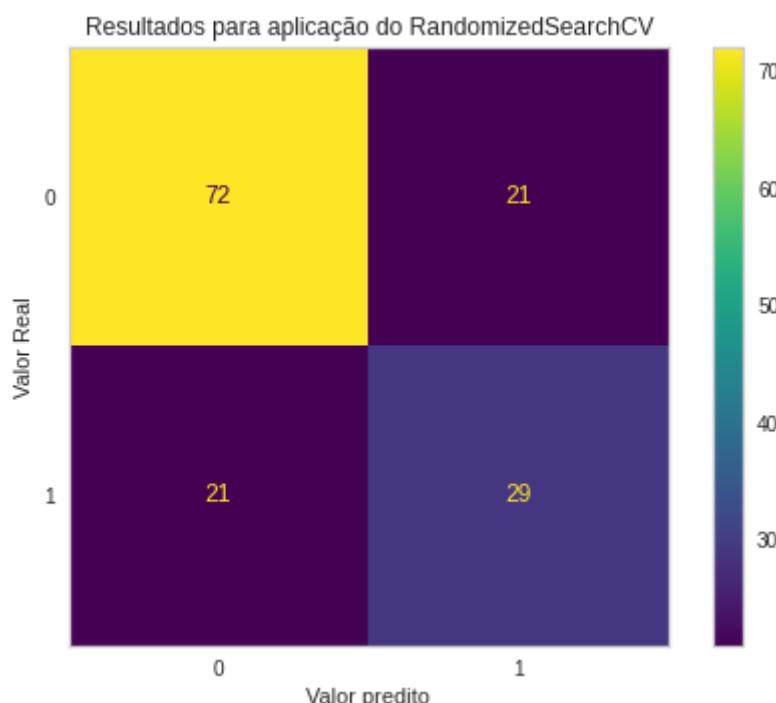


Figura 36: Matriz de confusão final da árvore de decisão

#### 4.5.5.4 Support Vector Machine

##### 4.5.5.4.1 Modelo default

Ao aplicar o Algoritmo Support Vector Machine, considerando as Features utilizadas, e os dados separados entre treino e teste, ele gerou um resultado de Acurácia (treino): 97% e de Acurácia (teste): 77%.

	Precision	Recall	f1-score	support
0	0.68	0.96	0.80	81
1	0.95	0.60	0.73	90
Accuracy			0.77	171
Macro avg	0.82	0.78	0.77	171
Weighted avg	0.82	0.77	0.77	171

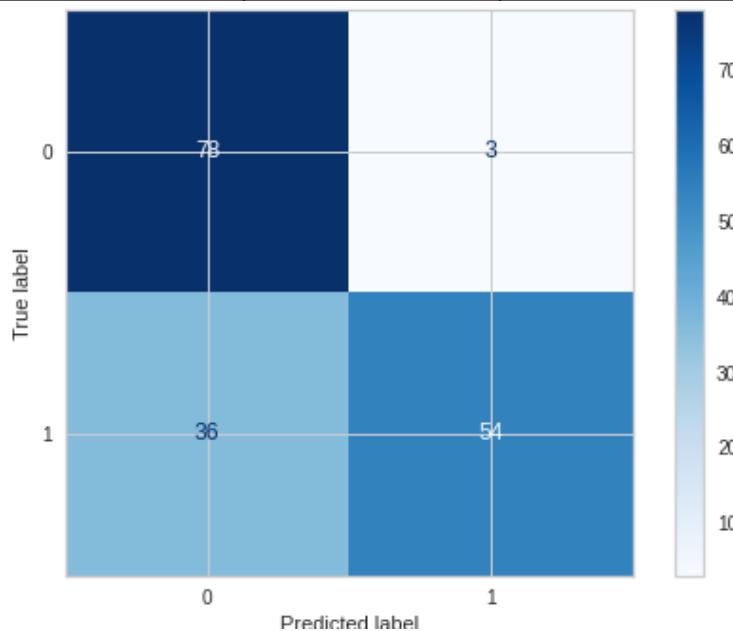


Figura 37. Matriz de confusão inicial para o Support Vector Machine

Ao observar a matriz de confusão, o modelo acertou 78 de 81 pessoas que estão ativas e acertou 54 de 90 que foram desligados da empresa.

##### 4.5.5.4.2 Aplicação e definição dos hiperparâmetros

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). Porém com o SVM, a aplicação e processamento dos

hiperparâmetros tornavam o modelo lento demais para ser calculado e foi inviável utilizar os hiperparâmetros.

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

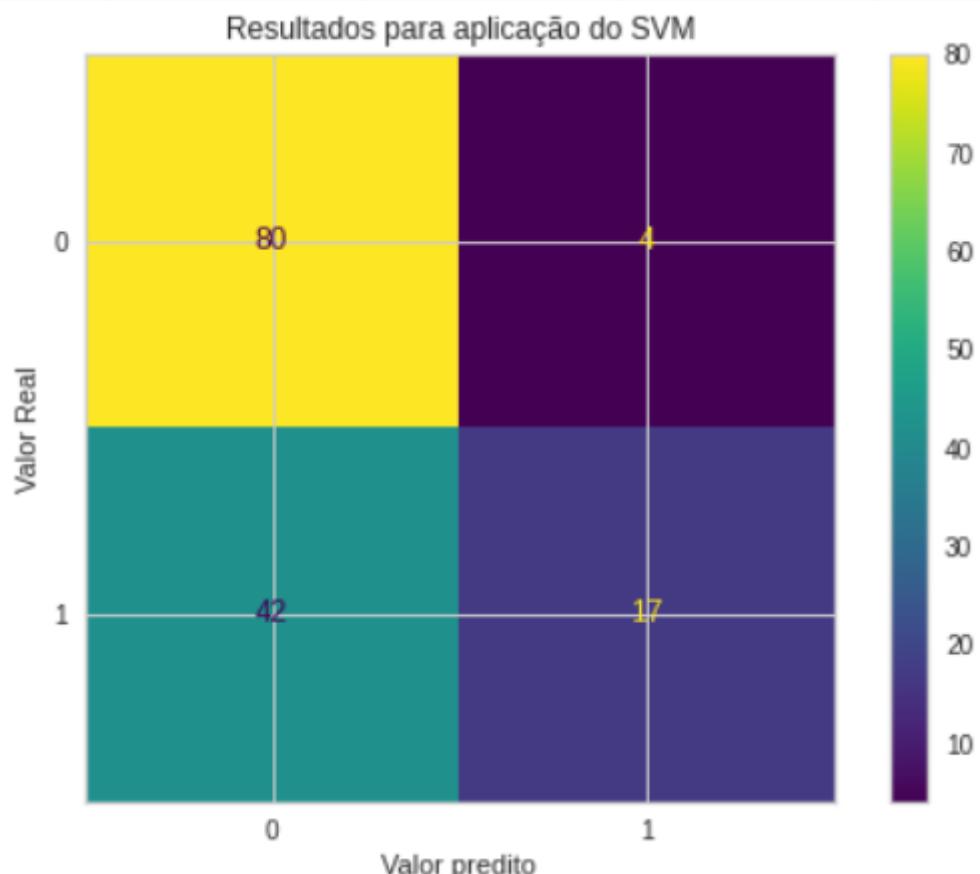


Figura 38. Matriz de confusão com hiperparâmetros para o Support Vector Machine

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 frequênci 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

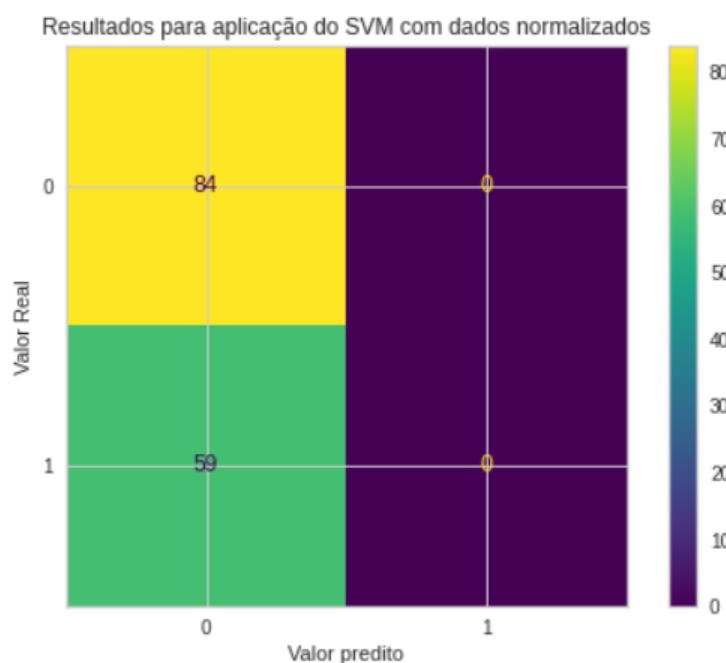


Figura 39. Matriz de confusão com normalização para o Support Vector Machine

	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

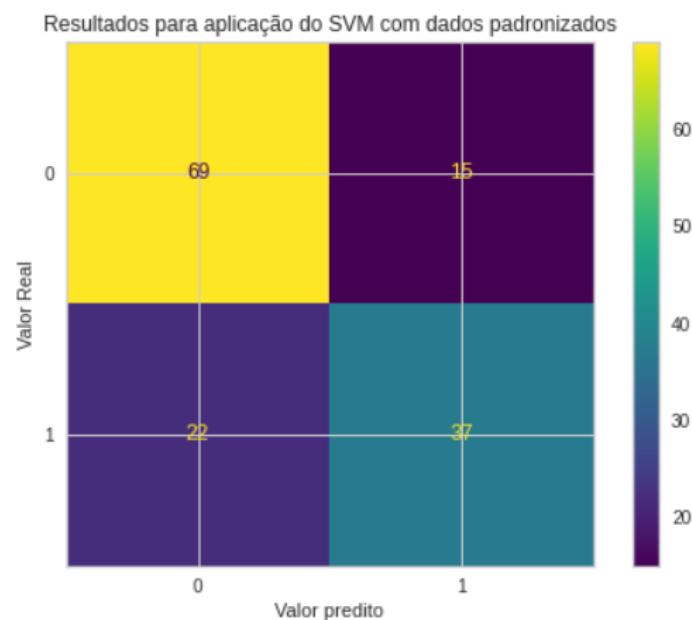


Figura 40. Matriz de confusão com padronização para o Support Vector Machine

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 acurárias.

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

Resultados para aplicação do SVM com RandomizedSearchCV

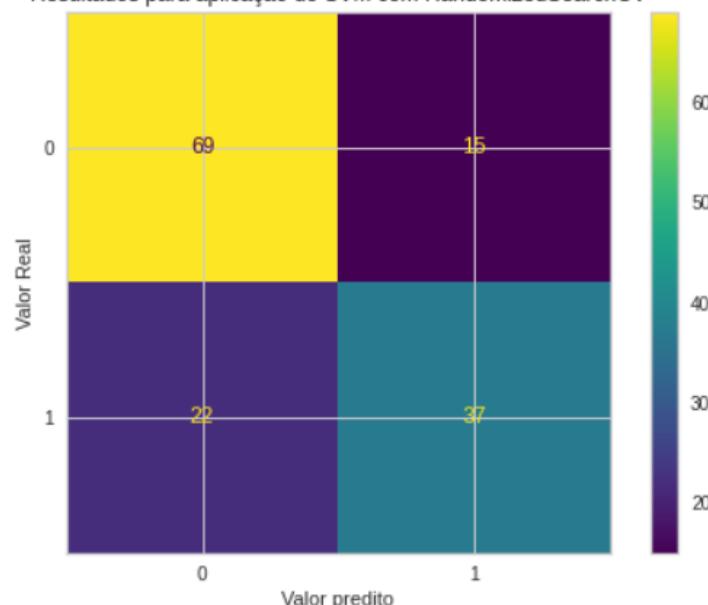


Figura 41. Matriz de confusão com Random Search no Support Vector Machine

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

#### 4.5.5.5 Random Forest

##### 4.5.5.5.1 Modelo default

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.

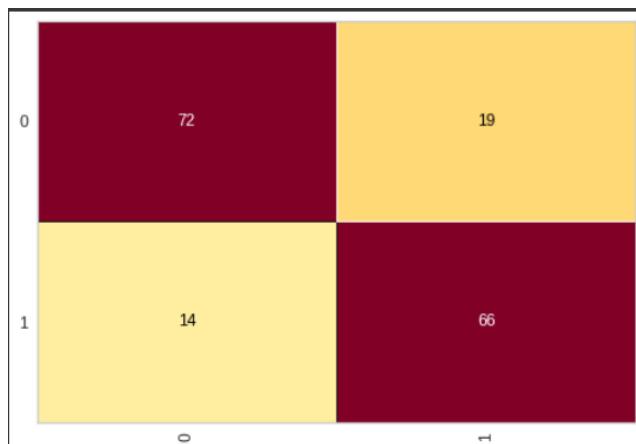


Figura 42. Matriz de confusão para o random forest

Observando a matriz de confusão do modelo Random Forest, pode-se concluir que a acurácia foi relativamente alta, uma vez que 72 das 91 previsões de permanência e 66 das 80 de saída da empresa estão corretas, ou seja, 14 das previsões são falsos negativos e 19 são falsos positivos.

	Precision	Recall	f1-score	support
0	0.84	0.79	0.81	91
1	0.78	0.82	0.80	80
Accuracy			0.81	171
Macro avg	0.81	0.81	0.81	171

Weighted avg	0.81	0.81	0.81	171
--------------	------	------	------	-----

Resultados das métricas de avaliação do Random Forest

Acuracidade (teste)	0.8070175438596491
---------------------	--------------------

#### 4.5.5.2 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 e o melhor resultado obtido foi com o Random Search, utilizando os seguintes hiperparâmetros: '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] e 'bootstrap': [True, False].

A acurácia obtida foi de 0.8128654970760234%, aumentando em comparação ao modelo com uso dos parâmetros default.

- Random Search

Parâmetros	Valor
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

bootstrap	True, False
-----------	-------------

#### 4.5.5.3 Variância de erro do modelo

No treinamento do modelo, é pedido um parâmetro chamado "random state". O modelo foi testado com números em um intervalo de 0 a 9. 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 Random Forest é de 14%. Segue um exemplo de 5 intervalos abaixo:

Acuracidade (teste): 0	0.75
Acuracidade (teste): 2	0.85
Acuracidade (teste): 4	0.81
Acuracidade (teste): 6	0.77
Acuracidade (teste): 7	0.86

Assim, conclui-se que o melhor valor para o random state neste caso é 7, que resulta em uma acurácia de 86%.

#### 4.5.5.6 Regressão Logística

##### 4.5.5.6.1 Modelo default

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, devido a discrepância do atributo 'Salário Mês'.

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:**

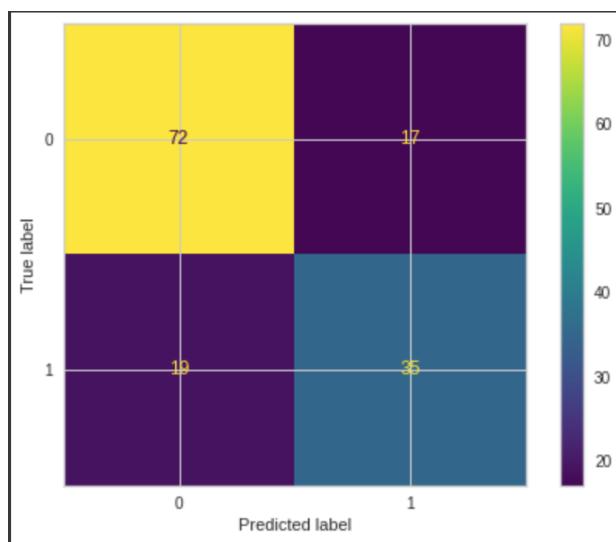


Figura 43. Matriz de confusão inicial para a regressão logística

	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.

#### 4.5.5.6.2 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:**

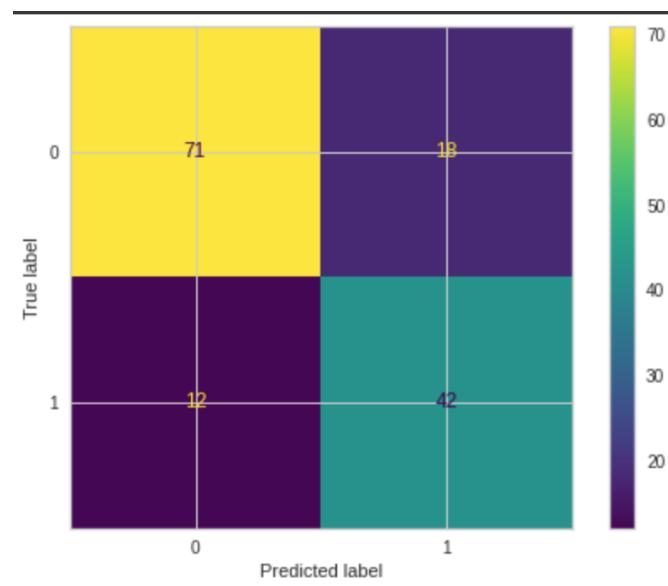


Figura 44. Matriz de confusão final para a regressão logística

	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.

#### 4.5.5.6.3 Variância do modelo

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

## 4.5.6. Conclusões dos modelos

### 4.5.6.1 Curva ROC

Para a confecção da curva ROC, aplicamos conceitos que predizem probabilidades de classificações em todos os métodos e colocamos os pontos de formato (Sensibilidade, Especificidade) num plano cartesiano. Com isso, construímos as curvas e calculamos a área sob elas.

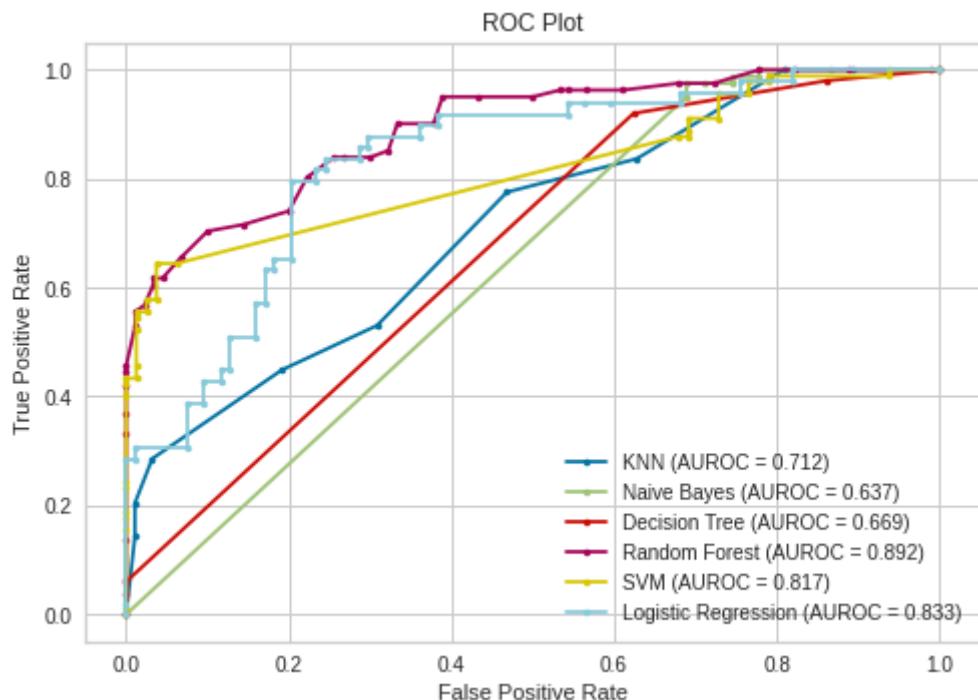


Figura 45. Visualização da curva Roc de todos os modelos

Segundo os resultados apresentados pela curva ROC, o modelo que melhor performa sem a aplicação dos hiperparâmetros é o Random Forest, com uma pontuação AUROC de 89.2% e a pior é o método Naive Bayes, com pontuação AUROC de 63.7%.

Com a aplicação de hiperparâmetros no modelo, os resultados são apresentados a seguir:

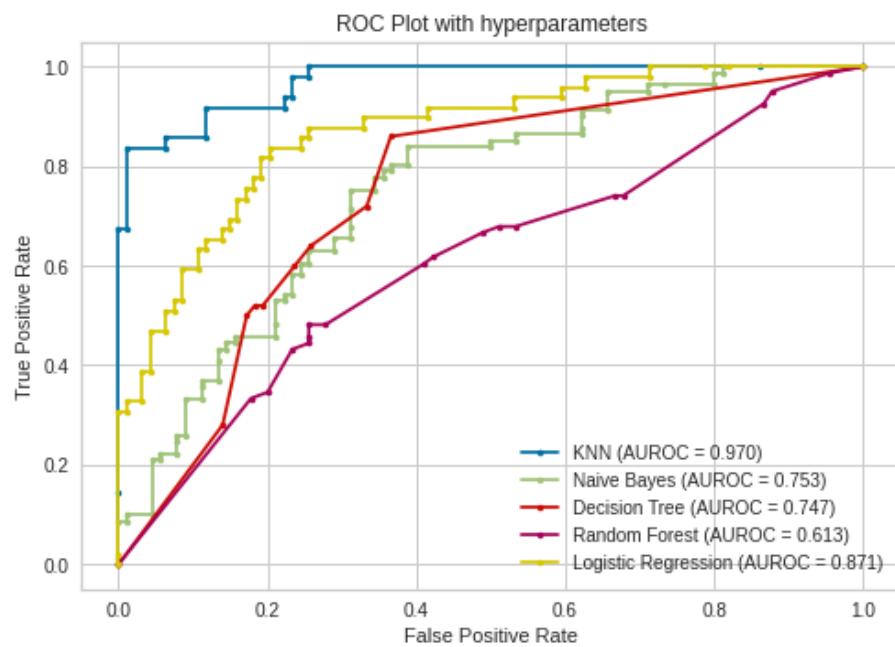


Figura 46. Visualização da curva roc de todos os modelos com hiperparâmetros

Com a aplicação dos hiperparâmetros, é notável o aumento na performance dos métodos, sendo que as maiores pontuações AUROC são de 97.0% com o KNN e 87.1% com a Regressão Logística e a menor foi 61.3% com Random Forest. Embora a pontuação do KNN seja alta, sua acurácia é baixa comparada às dos demais modelos. A curva do modelo SVM foi, a princípio, retirada do gráfico, pois a aplicação e processamento dos hiperparâmetros tornavam o modelo lento demais para ser calculado.

Analizando os resultados de todos os seis modelos, chegou-se à conclusão de que o modelo de maior qualidade é o da Regressão Linear, uma vez que apresenta os maiores índices dentre as avaliações conjuntas.

Após testar muitas vezes o mesmo modelo com vários atributos diferentes, chegamos no resultado final de 81% de acurácia utilizando os melhores parâmetros encontrados para o algoritmo de regressão linear 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)	0.8132530120481928
Acuracidade (teste)	0.8041958041958042

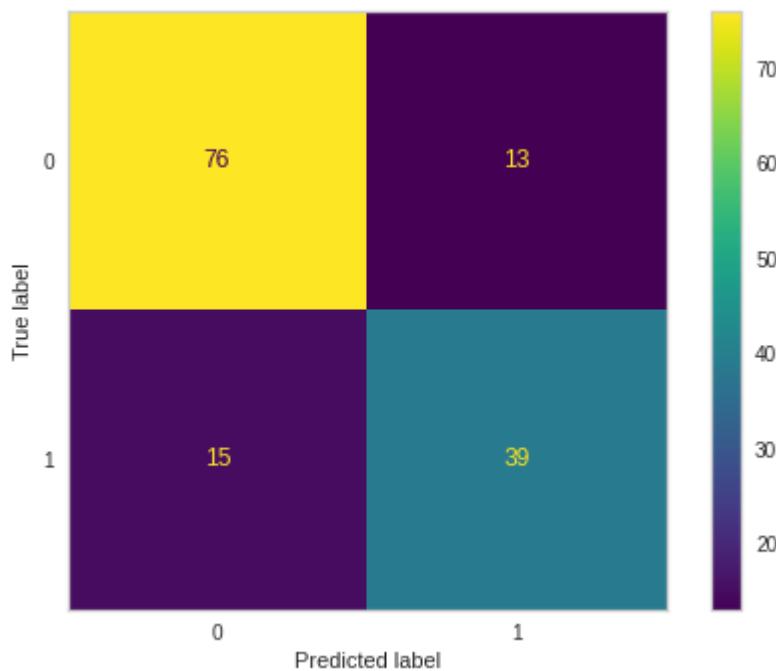


Figura 47. Matriz de confusão final do modelo escolhido

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

Após todas as análises dos modelos e das pontuações geradas, podemos concluir que os resultados, principalmente aqueles de melhor performance resultantes da regressão logística e da Random Forest, possuem uma confiabilidade relativamente grande (cerca de 80% de acurácia) e podem ser levados em consideração para otimizar a tomada de decisão. Porém é importante ressaltar que as previsões feitas não devem ser tratadas como resultados absolutos, cabendo aos usuários do modelo a tomada de decisão final.

Para a utilização do modelo, recomendamos que as bases de dados utilizadas tenham um formato similar ao da fornecida para a confecção dos modelos (tipo de arquivo e formato dos dados). Para o tratamento dos dados e derivação de novos atributos, foram utilizados os nomes das colunas que representam variáveis diferentes. Deve-se, portanto, atentar-se a nomes diferentes dos originalmente estabelecidos para que o software consiga extrair as informações necessárias corretamente do documento importado. Caso seja necessário alterar o nome de alguma coluna, deve-se fazer uma revisão no código-fonte e adequá-lo à nova nomenclatura a fim de evitar erros.

Por fim, em relação à análises feitas pelos modelos, recomendamos a utilização dos resultados da regressão logística para análises mais assertivas, porém os demais também podem ser consultados como segunda via.

## 6. Referências

CRISP-DM Help Overview. [S. I.]: © 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. I.]: 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. I.]: © 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. I.]: 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. I.], 28 jul. 2019. Disponível em:

<https://medium.com/turing-talks/turing-talks-12-classificacao-por-svm-f4598094a3f1>. Acesso em: 9 set. 2022.

1.4. Support Vector Machines. Scikit-learn 1.1.2. [S. I.]: © 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.

COUTINHO, B. Disponível em:

⟨<https://medium.com/turing-talks/turing-talks-12-classificacao-por-svm-f4598094a3f1>⟩.

Acesso em: 6 out. 2022.

Salesforce quer mais parceiros. Disponível em:

⟨<https://www.baguete.com.br/noticias/27/06/2022/salesforce-quer-mais-parceiros>⟩. Acesso em: 6 out. 2022.

Low-Code/No Code. Disponível em:

⟨<https://imasters.com.br/noticia/novo-estudo-sobre-mercado-salesforce-reforca-avanco-do-movimento-low-code-no-code>⟩. Acesso em: 6 out. 2022.

Tecnologia em nuvem: veja as tendências para 2022! Disponível em:

⟨<https://santodigital.com.br/5-tendencias-de-tecnologia-em-nuvem-para-2021/>⟩. Acesso em: 6 out. 2022.

GARCIA, Ana Cristina Bicharra. Ética e inteligência artificial. Computação Brasil, [S. I.], p. 14-22, 1 nov. 2020. Disponível em:

<https://sol.sbc.org.br/journals/index.php/comp-br/article/view/1791/1625>.  
Acesso em: 5 out. 2022.

Régressão Logística. Disponível em:

⟨<https://matheusfacure.github.io/2017/02/25/regr-log/>⟩.

REGRESSÃO LOGÍSTICA. [s.l: s.n.]. Disponível em:

⟨[https://edisciplinas.usp.br/pluginfile.php/3769787/mod\\_resource/content/1/09\\_RegressaoLogistica.pdf](https://edisciplinas.usp.br/pluginfile.php/3769787/mod_resource/content/1/09_RegressaoLogistica.pdf)⟩.

Everymind – Líder no ecossistema Salesforce para o Brasil pelo segundo ano consecutivo.

Disponível em: ⟨<https://www.everymind.com.br>⟩. Acesso em: 6 out. 2022.

População mais velha não para de crescer e demora mais a sair do mercado. Disponível em:

⟨<https://veja.abril.com.br/economia/populacao-mais-velha-nao-parade-crescer-e-demora-mais-a-sair-do-mercado/>⟩. Acesso em: 6 out. 2022.