RAPPREDICTION
RAPPI

Controle do Documento

Histórico de revisões

Data	Autor	Versão	Resumo da atividade
08/08/2022	Pablo Ruan e Leandro Custódio	1.1	Criação do documento
09/08/2022	Felipe Sampaio e Leandro Custódio	1.2	2.2) Análise do Cenário: Matriz SWOT
10/08/2022	Felipe Sampaio e Pedro Silva	1.3	2.3) Proposta de Valor: Value Proposition
11/08/2022	Felipe Sampaio e Pedro Silva	1.4	2.1) Objetivos
11/08/2022	Pedro Silva e Rafael Moritz	1.5	2.5) Matriz de Risco
11/08/2022	Leandro Custódio e Pablo Viana	1.6	4.2) Compreensão de dados
11/08/2022	Gabriel Pascoli	1.7	4.1.6)Base da Persona
11/08/2022	Felipe Sampaio	1.8	4.1.3) Planejamento Geral da Solução
11/08/2022	Felipe Saadi	1.9	4.1.1) Contexto da Indústria
12/08/2022	Felipe Saadi	2.0	4.1.6) Mapa dos Stakeholders 4.1.7) Matriz Poder x Interesse
25/08/2022	Leandro Custódio	2.1	4.2) adição dos campos supply e tempo resolução e modal-002 em Compreensão dos Dados
25/08/2022	Felipe Sampaio	2.2	4.1.6) Finalização da Persona
25/08/2022	Felipe Saadi e Gabriel Pascoli	2.3	4.1.9) Jornada do Usuário
25/08/2022	Felipe Saadi	2.4	4.1.6) Mapa dos Stakeholders
28/08/2022	Felipe Sampaio Felipe Saadi	2.5	4.3) Preparação dos dados
10/09/2022	Felipe Sampaio Rafael Moritz	3.0	4.4) Modelagem
25/09/2022	Felipe Sampaio Felipe Saadi Rafael Moritz	4.0	4.4) Continuação da seção Modelagem 4.5) Avaliação
17/03/2023	Escritório de Projetos	5.0	Protegendo dados sensíveis

Sumário

4.2) adição dos campos supply e tempo resolução e modal-002 em Compreensão dos Dados 2

1. Introdução 4

2. Objetivos e Justificativa 4

2.1. Objetivos 4

2.2. Justificativa 5

3. Metodologia 6

3.1. CRISP-DM 7

3.2. Ferramentas 8

3.3. Principais técnicas empregadas 9

4. Desenvolvimento e Resultados 10

4.1. Compreensão do Problema 10

4.1.1. Contexto da indústria 10

4.1.2. Análise SWOT 11

4.1.3. Planejamento Geral da Solução 13

4.1.4. Value Proposition Canvas 14

4.1.5. Matriz de Riscos 16

4.1.7. Matriz Poder x Interesse 17

4.1.8. Personas 17

4.1.9. Jornadas do Usuário 19

4.2. Compreensão dos Dados 19

4.3. Preparação dos Dados 20

4.4. Modelagem 24

4.5. Avaliação 33

4.6 Comparação de Modelos 34

5. Conclusões e Recomendações 37

6. Referências 40

Anexos 41

1. Introdução

A Rappi é uma empresa de tecnologia que proporciona o desenvolvimento do comércio em diversas cidades através da entrega de produtos por uma plataforma digital própria. Seu aplicativo de delivery cria uma rede de contato ampla que atua entre consumidores, entregadores, parceiros e fornecedores (estabelecimentos em geral), e ópera no setor de transporte de produtos alimentícios, farmacêuticos, uso geral, etc.

É uma instituição com vários locais de atuação, dentre elas, as principais capitais do Brasil e sedes em outros lugares do mundo, como Colômbia, México, Argentina, etc. Apesar de não possuir a maior influência do mercado dentro desse ramo, possui um grande nicho de mercado e possui diversos setores de atuação dentro do delivery.

A organização da Rappi é formada por um time de operações, com profissionais contratados pela empresa, responsável pela administração geral dos setores. A outra parte é formada pelos RappiTenderos (terceirizados), responsáveis por receber o pedido e realizar as entregas para os usuários finais. Nesse sentido, a solução desejada pela Rappi consiste em um sistema, cuja finalidade é apresentar qual a tendência de churn de cada entregador através de uma escala que varia entre 1 e 5, conforme a probabilidade.

2. Objetivos e Justificativa

2.1. Objetivos

Problema a ser resolvido:

O problema apresentado pela Rappi é o alto número de saída dos entregadores (churn), visto que eles consideram a saída de um Rap piTendero (RT) a partir de um período de 21 dias de inatividade.

Dados disponíveis:

Os dados fornecidos pela Rappi vieram como tabelas em formato csv, contendo diversos tipos de informações, tais como:

- Informações de ganho dos RT’s;
- Número de pedidos cancelados;
- Quantidade de RT’s em determinados níveis (diamond, silver, etc.);
- Quantidade de RT’s do gênero feminino e masculino;
- Quantidade de RT 's ativos.

Solução proposta:

A solução final consiste em um algoritmo de predição treinado com dados fornecidos pelo cliente, a aplicação terá uma resposta binária (true, false) com relação a um RappiTendero dar churn (inatividade por mais de 21 dias na plataforma), a saída será mostrada em uma tela final, com um sim ou não.

Tipo de tarefa (regressão ou classificação):

A tarefa terá caráter de classificação: ela mostrará eventos que podem acontecer, se não for um, será o outro.

Solução Proposta utilizada:

O sistema será utilizado pelo time de operações da própria Rappi. Ela será utilizada dentro do colab, importando o arquivo csv e rodando as células de execução.

Benefícios trazidos pela solução:

A solução trará benefícios para Rappi, pois ela conseguirá tomar medidas estratégicas em relação à possibilidade de saída dos entregadores, assim como para os entregadores que terão uma visibilidade maior dentro da empresa.

Critério de sucesso:

A fase de teste do modelo preditivo será dividida em 3 etapas:

1.º) A primeira fase será a alimentação da IA, com cerca de 70% dos dados, nesse momento, ela será treinada para realizar as análises, porém há o risco do modelo se tornar enviesado por se adaptar, exclusivamente, aos dados inseridos.
2.º) A segunda fase será o teste da IA, com cerca de 20% dos dados, nesse momento, ela será testada como um modo de verificar se há algum vício ou tendência por parte do modelo.
3.º) A terceira fase será a validação da IA, com cerca de 10% dos dados, nesse momento, ela passará por uma última verificação antes de ser lançada para o cliente final.

O critério de sucesso do grupo será manter uma coerência em relação às 3 fases de teste da IA, será feita uma comparação em relação aos dados apresentados durante as etapas teste e o nível de precisão em relação aos dados apresentados, através do cálculo da média, desvio padrão em relação a cada teste.

2.2. Justificativa

O impacto trazido pela solução final oferece um poder de escolha para a Rappi, pois a partir dos resultados fornecidos pelas predições, a empresa poderá optar pelas próximas estratégias de negócios em diferentes cenários do mercado.

O modelo oferece uma análise sobre aqueles entregadores com possibilidade de dar churn da empresa com uma alta precisão, sendo assim, a Rappi consegue aprimorar sua organização em relação à saída de entregadores, se preparando para situações futuras.

A solução possui um diferencial em relação a sua simples usabilidade e alta taxa de acerto dentro do objetivo principal, que no caso é encontrar os RT's com maior probabilidade de dar churn, possibilitando uma economia de recursos e tempo, visto que é a Rappi pode se preparar para uma saída em massa de entregadores ou para um momento de recessão econômica.

A entrega final do modelo é em uma interface bem simples de uso, com o código oculto e apresentando apenas a funcionalidade daquela célula, isso possibilita uma facilidade de uso por grande parte do time de operações e o entendimento dos resultados obtidos.

3. Metodologia

O modelo de organização utilizado no grupo foi a Metodologia Ágil "Scrum", em que o desenvolvimento do projeto é dividido em 5 Sprints, com duração de 2 semanas cada. Abaixo há uma descrição geral do desenvolvimento do grupo em cada etapa em conjunto com a metodologia CRISP-DM:

Sprint 1:

- Primeiro contato em reunião com o cliente;
- Entendimento do problema trazido pela empresa (Como vamos resolver?);
- Análises de negócios (Matriz SWOT, Canvas Value Proposition, Matriz de Risco, etc.);
- Análise dos dados fornecidos pela empresa e se havia correlação entre as informações fornecidas e o problema a ser solucionado;
- Descrição dos principais itens dos dataframes disponibilizados (significados das colunas, linhas e das tabelas);
- Elaboração de hipóteses;
- Criação de gráficos para suportar as hipóteses.

Sprint 2:

- Limpeza e filtragem dos dados recebidos;
- Análise dos StakeHolders envolvidos com a Rappi;
- Análise do usuário final da solução (persona);
- Elaboração e implementação de features;
- Início da unificação das tabelas utilizadas.

Sprint 3:

- Retratamento dos dados fornecidos pelo cliente;
- Elaboração e implementação de novas features;
- Teste com 8 modelos preditivos escolhidos pelo grupo;
- Análise e descrição dos resultados obtidos.

Sprint 4:

- Teste dos hiperparâmetros;
- Utilização da ferramenta Pycaret para encontrar os melhores modelos de revocação;
- Análise e comparação dos modelos.

3.1. CRISP-DM

Esse tópico aborda a metodologia aplicada no processo de mineração de dados, cuja finalidade é transformar dados da empresa em informações e conhecimentos utilizados para planejamentos estratégicos. Sua aplicação está relacionada à matemática e estatística para realização do cruzamento de dados e essas manipulações com alto volume de dados que auxiliam muito na tomada de decisões para a empresa. O grupo optou por utilizar esse método de análise pela sua eficiência e flexibilidade, junto às metodologias ágeis, visto que há sempre uma validação de entregas por sprint (etapas determinadas).

Como funciona o CRISP-DM:

O CRISP-DM é uma metodologia cíclica que tende a se repetir ao longo das Sprints, então durante todo o desenvolvimento, há a possibilidade de realizar uma nova análise acerca dos dados fornecidos para o time. Abaixo segue o resumo do modelo CRISP-DM e suas etapas:

- **Entendimento do problema**: compreender qual o real problema da empresa e qual o impacto gerado?
- **Compreensão de dados:** documentar e descrever os dados disponíveis, sendo também o início da mineração de dados, com a separação das informações mais relevantes.
- **Preparação dos dados:** Fase técnica de separação de dados e data frames, estruturação, etc.
- **Modelagem:** Momento em que são realizadas as predições de negócio, teste de modelos preditivos.
- **Avaliação:** Checagem dos resultados, a partir das predições realizadas.
- **Implementação dos modelos na empresa:** O resultado final é aplicado dentro da empresa e tem um grande impacto.

Vantagens:

- **Bom relacionamento com o cliente**, pois o problema é debatido com o cliente para entender suas necessidades ao longo do processo de desenvolvimento.
- **Orientação no momento de decisões**, visto que com análises e diversas predições tomadas há um gerenciamento de riscos muito mais consciente.
- **Aplicação de novos modelos para solução de outros problemas na empresa**, a partir da análise de dados é possível entender a situação atual da empresa e apresentar as melhores alternativas para ela.
- **Análises em tempo real do cenário interno e externo em tempo real**, a leitura dos dados e a alta interatividade com o cliente favorece trazer informações sobre o ambiente dentro e fora da empresa.

3.2. Ferramentas

A principal ferramenta utilizada pelo grupo foi o **Google Collaboratory (Colab)**. Essa é uma interface com serviço gratuito na nuvem do próprio Google. É uma plataforma muito útil para criação de Inteligências Artificiais e utilização de técnicas de Machine Learning. Foi o principal serviço utilizado, pois todos os códigos (modelos preditivos, gráficos, carregamento de data frames, etc) foram executados dentro desta plataforma.

Além disso, uma ferramenta que foi muito importante durante o desenvolvimento do projeto foi o **Github**. É um serviço de hospedagem na Internet para desenvolvimento de software e controle de versão. A cada Sprint o grupo realizou o upload da documentação e do modelo preditivo em um repositório privado, para manter uma versão atualizada e com alterações realizadas em um dado período.

Outra plataforma utilizada pelo grupo foi o **Google Drive**. Essa aplicação é essencial quando se trabalha com o Google Colab, uma vez que consegue conectar todos os arquivos do computador com o arquivo collaboratory, possibilitando salvar seus códigos e compartilhar com os outros integrantes do grupo.

Aliás, a ferramenta aplicada para criação e edição da documentação foi o **Google Docs**, dado que é uma plataforma completa com diversas funcionalidades e com alternativa de modificações simultâneas em grupo.

Por fim, o **Trello**. Essa plataforma permite a utilização de um quadro e a criação de cards com as tarefas que o grupo deve realizar. Com isso, é possível atribuir cada integrante para uma ou mais funções e depois movê-los para cada lista (A fazer, Pensar sobre, Em andamento, etc).

3.3. Principais técnicas empregadas

A seguir estão listados os principais métodos utilizados no desenvolvimento, dentre eles fontes de consulta para embasamento teórico, linguagem de programação utilizada, etc.

Python: Linguagem de programação utilizada para o projeto. Muito eficiente, simples e consistente, além de ter acesso a ótimas bibliotecas e estruturas para IA e Machine Learning.

Pandas: Biblioteca usada para análise e manipulação de dados.

Sk Learning: Biblioteca usada para Machine Learning.

Pycaret: Biblioteca usada para automatizar o processo de análise dos resultados durante o processo de Machine Learning.

Numpy: Biblioteca numérica usada para trabalhar com arrays.

MatplotLib: Biblioteca usada na visualização de dados para a criação de interfaces gráficas.

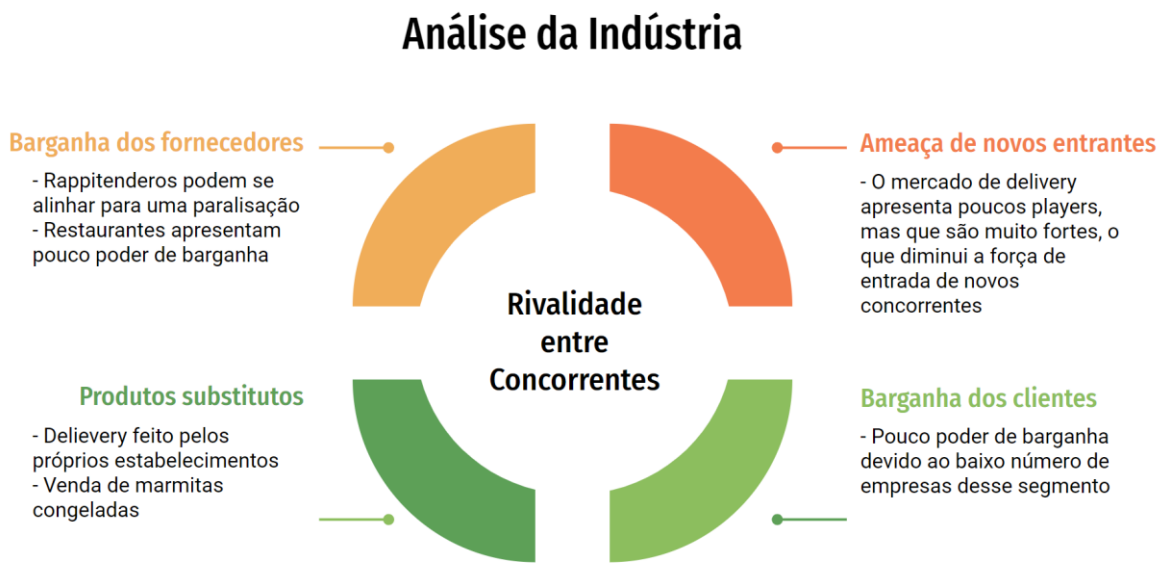
Stack overflow: Uma comunidade virtual usada para encontrar e contribuir com respostas para desafios técnicos, na maioria das vezes de programação.

4. Desenvolvimento e Resultados

4.1. Compreensão do Problema

4.1.1. Contexto da indústria

Consiste em um esquema de rede de abrangência e contato da Rappi com toda uma cadeia apresentada abaixo:



Concorrentes:

No ramo de entrega de alimentos, o principal adversário é o Ifood, que domina mais de 80% do delivery de restaurantes. Devido a tamanha dominância, um dos principais players do mercado, o Uber Eats, encerrou a sua atuação no Brasil em março, deixando a sua participação de mais de 10% do mercado. Em relação a entregas gerais, a empresa Loggi oferece serviços semelhantes a Rappi, mas apresenta uma menor fatia do mercado.

Modelo de Negócios:

O Rappi é uma empresa que tem como modelo de negócios um aplicativo multifuncional, que proporciona ao usuário a oportunidade de fazer praticamente qualquer pedido. Permite desde a compra de um produto de um estabelecimento parceiro a até a entrega de um objeto importante para outra pessoa, as opções de entrega são diversas. A empresa também ataca

outras frentes, como a de atuar como uma agência de viagens. Seu modelo de lucro é por meio de parcerias com estabelecimentos, que pagam uma porcentagem do valor recebido pelas compras feitas por meio da plataforma do Rappi. Além disso, possui um programa de assinatura pago cujo objetivo é fidelizar clientes na plataforma, entregando para eles alguns benefícios, como o frete grátis.

Tendências:

As principais tendências desse mercado são as **dark kitchens**, que permitem aos estabelecimentos a produção de produtos destinados especialmente para a entrega por deliverys, sendo capaz de suprir melhor as enormes demandas das plataformas de delivery; **entregas ágeis**, como as pessoas estão vivendo cada dia mais com pressa, é faz necessário diminuir o tempo de espera, o fator tempo é crucial para manter um nível de qualidade de uma empresa de delivery; **produtos saudáveis**, por conta do aumento de problemas de saúde, as pessoas têm buscado cada vez mais alimentos benéficos para a saúde, exigindo que os aplicativos de delivery aumentem a diversificação de produtos em suas plataformas; **venda em multicanais**, com o uso da tecnologia por meio de diferentes dispositivos durante o dia a dia de usuários, há uma necessidade das plataformas de delivery proporcionarem aos seus usuários múltiplos canais para fazerem pedidos.

4.1.2. Análise SWOT

A análise SWOT é uma ferramenta de gestão que serve para fazer o planejamento estratégico de empresas e novos projetos. A sigla SWOT significa: **Strengths** (Forças), **Weaknesses** (Fraquezas), **Opportunities** (Oportunidades) e **Threats** (Ameaças).

As duas linhas de cima (Forças e Fraquezas) mostram características internas da Rappi, já as duas linhas de baixo (Oportunidades e Ameaças) analisam o contexto externo da Rappi, mas influenciam diretamente a empresa.

Forças:

- Alta abrangência territorial na América Latina.
- Incentiva o empreendedorismo na base de funcionários.
- Entregas eficientes.
- Altíssima variedade no tipo de produtos entregues “quase qualquer coisa”.
- Horário de funcionamento 24 horas.
- Interface eficiente.
- Potencial de crescimento.
- Empresa inovadora.

Fraquezas:

- A empresa não oferece segurança financeira para entregadores.
- Muitas reclamações de Rts referentes ao sistema de banimento.
- Muitas reclamações acerca do modo Turbo do Rappi.
- Entregadores insatisfeitos com as condições de trabalho.

Oportunidades:

- Diferencial da implementação de bicicletas elétricas espalhadas pelo Brasil.
- Aumento na demanda de pedidos em aplicativos de delivery.
- Poucos concorrentes em algumas regiões.
- Entregas automatizadas (drones, robôs)
- Aprimoramento do RappiPay*.

Ameaças:

- Concorrência com alto domínio no mercado.
- Cancelamento de pedido por diversos motivos.
- Alta volatilidade no número de Rts.
- Problemas relacionados à legislação federal.

4.1.3. Planejamento Geral da Solução

- Problema a ser resolvido:

O problema apresentado pela Rappi é o alto número de saída dos entregadores (churn), visto que eles consideram a saída de um RappiTendero (RT) a partir de um período com inatividade de 21 dias.

- Dados disponíveis:

Os dados fornecidos pela Rappi vieram como tabelas em formato csv, contendo diversos tipos de informações, tais como:

  - Informações de ganho dos RT's;
  - Número de pedidos cancelados;
  - Quantidade de RT's em determinados níveis (diamond, silver, etc);
  - Quantidade de RT's do gênero feminino e masculino;
  - Quantidade de RT's ativos.

- Solução proposta:

A solução final consiste em um algoritmo de predição treinado com dados fornecidos pelo cliente, a aplicação terá um resposta binária (true, false) com relação a um RappiTendero dar churn (inatividade por mais de 21 dias na plataforma), a saída será mostrada em uma tela final, com um sim ou não.

- Tipo de tarefa (regressão ou classificação):

A tarefa terá caráter de classificação, pois ela mostrará eventos que podem acontecer, se não for um será outro

- Funcionamento da solução proposta:

O sistema será utilizado pelo time de operações da própria Rappi. Ela será utilizada dentro do colab, importando o arquivo csv e rodando as células de execução.

- Benefícios trazidos pela solução proposta:

A solução trará benefícios para Rappi, pois ela conseguirá tomar medidas estratégicas em relação a possibilidade de saída dos entregadores, assim como para os entregadores que terão uma visibilidade maior dentro da empresa.

- Critério de sucesso e a medida utilizada para avaliar:

A fase de teste do modelo preditivo será dividida em 3 etapas:

1°) A primeira fase será a alimentação da IA, com cerca de 70% dos dados, nesse momento, ela será treinada para realizar as análises, porém há o risco do modelo se tornar enviesado por se adaptar, exclusivamente, aos dados inseridos.

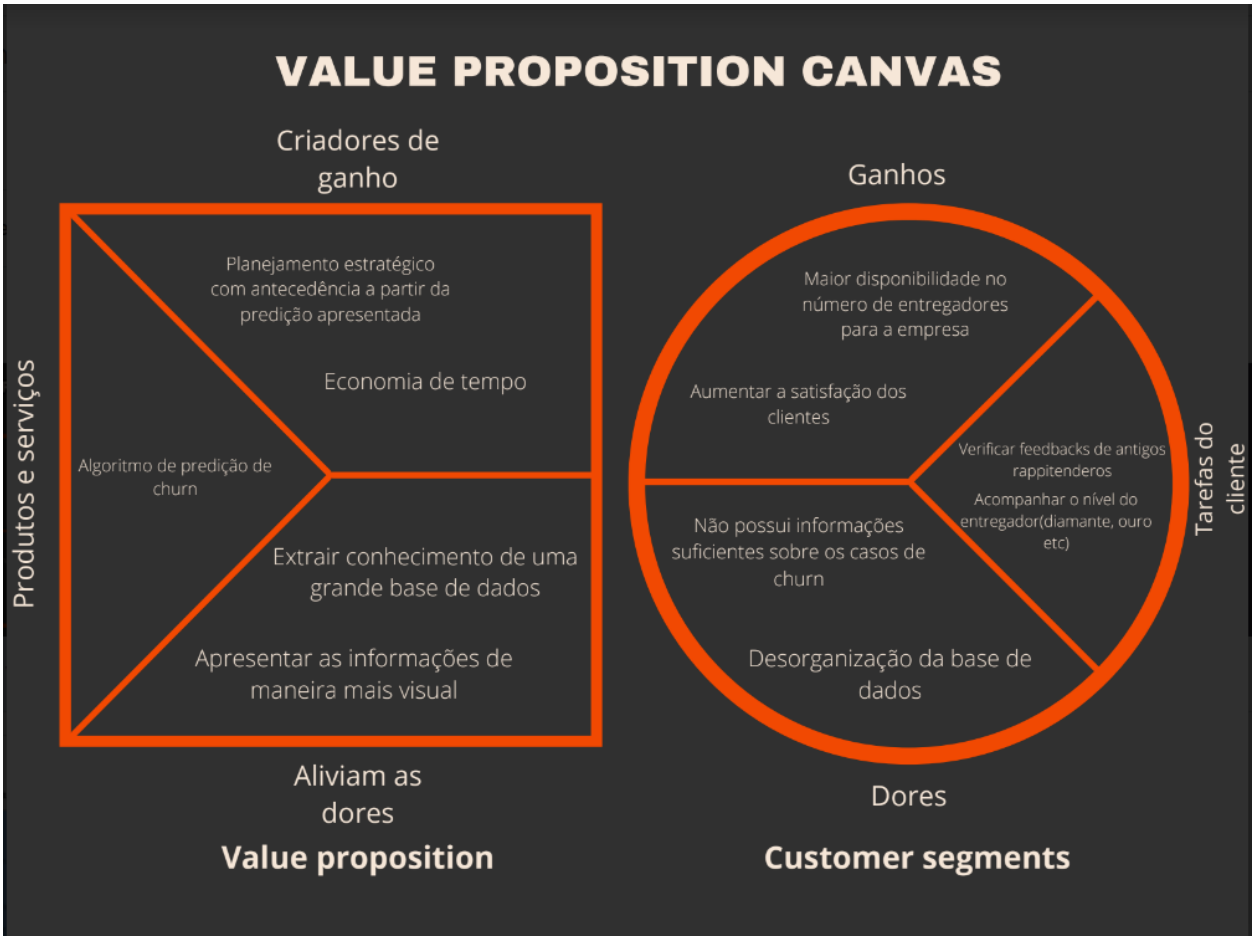
2°) A segunda fase será o teste da IA, com cerca de 20% dos dados, nesse momento, ela será testada como um modo de verificar se há algum vício ou tendência por parte do modelo.

3°) A terceira fase será a validação da IA, com cerca de 10% dos dados, nesse momento, ela passará por uma última verificação antes de ser lançada para o cliente final.

O critério de sucesso do grupo será manter uma coerência em relação às 3 fases de teste da IA, será feita uma comparação em relação aos dados apresentados durante as etapas teste e o nível de precisão em relação aos dados apresentados, através do cálculo da média, desvio padrão em relação a cada teste.

4.1.4. Value Proposition Canvas

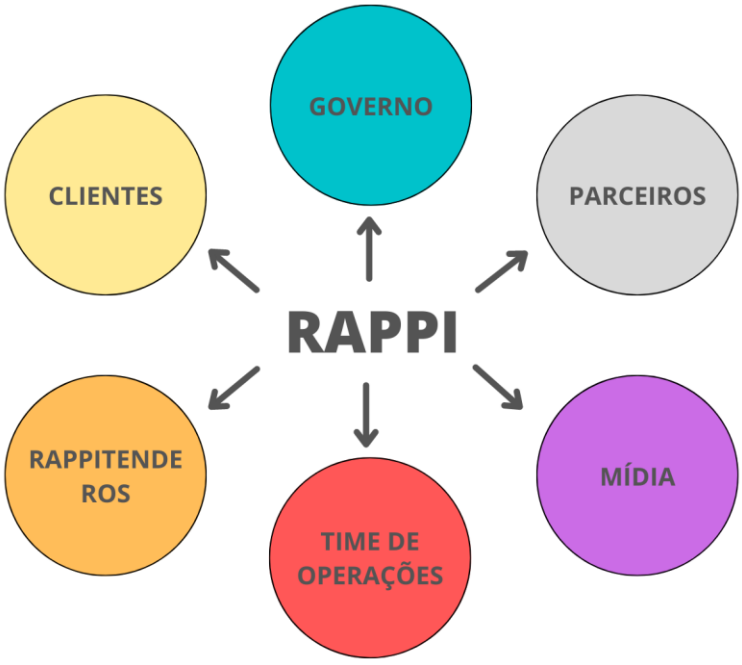
O Canvas de valor faz uma relação entre a necessidade do parceiro (dores, ganhos e tarefas) com o que será desenvolvido na nossa solução, ou seja, é feita uma análise para validar se o que estamos desenvolvendo está realmente alinhado com o que o cliente precisa.


VALUE PROPOSITION CANVAS
Criadores de ganho
Planejamento estratégico com antecedência a partir da predição apresentada
Economia de tempo
Algoritmo de predição de churn
Extrair conhecimento de uma grande base de dados
Apresentar as informações de maneira mais visual
Produtos e serviços
Aliviam as dores
Value proposition
Ganhos
Maior disponibilidade no número de entregadores para a empresa
Aumentar a satisfação dos clientes
Verificar feedbacks de antigos rappitenderos
Acompanhar o nível do entregador(diamante, ouro etc)
Não possui informações suficientes sobre os casos de churn
Desorganização da base de dados
Tarefas do cliente
Dores
Customer segments

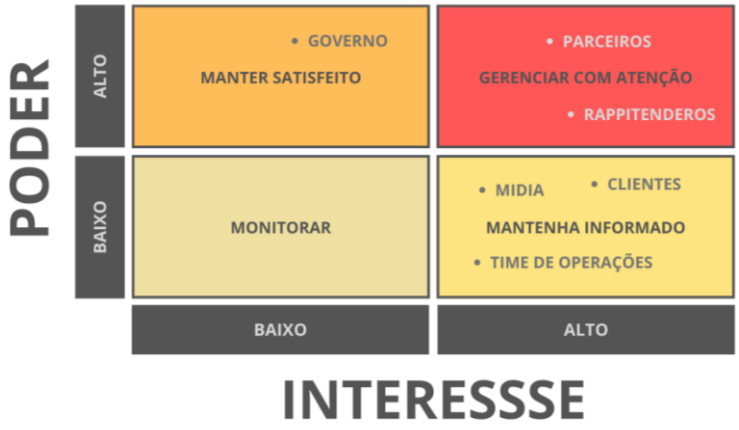
4.1.5. Matriz de Riscos

Matriz de Risco											
Probabilidade		Ameaças					Oportunidade				
Muito Alta	5							Temos um time muito forte e esforçado	Harmonia entre participantes do grupo		
Alta	4				O escopo do projeto não ser tão viável como o cliente imagina	Imprecisão nos outputs		Conhecimento prévio na linguagem de programação Python			
Médio	3			Conflito de ideias		Não entregarmos um produto no nível da expectativa do cliente por não termos familiaridade com AI		Professores possuem experiência com Inteligência Artificial			
Baixa	2		Problema com os dados								
Muito Baixa	1	Ficarmos muito tempo conversando ao invés de trabalhar, e acabar entregando um produto abaixo do nosso potêncial									
		1	2	3	4	5	5	4	3	2	1
		Muito Baixo	Baixo	Médio	Alta	Muito Alta	Muito Alta	Alta	Médio	Baixo	Muito Baixo
		Impacto									

4.1.6. Mapa dos Stakeholders



4.1.7. Matriz Poder x Interesse



4.1.8. Personas

Rappitendero:

Nome: Jaime
Idade: 37 anos
Gênero: Masculino
Ocupação: Entregador

Bio:

James é um jovem casado e possui 2 filhos para sustentar, trabalha como RappiTendero há cerca de 1 ano e o valor das entregas junto as gorjetas são suas fontes de renda. Possui o Ensino Superior Completo e está aberto a novas oportunidades de trabalho em sua área de graduação, aprecia momentos de lazer com a família, tais como jogar futebol com as crianças e sair com sua esposa. Trabalha na Rappi pela flexibilidade de horários, então consegue se organizar bem com o tempo em que vai trabalhar e consegue fazer uma faculdade EaD.

Principais Dores:

- Punições por motivos não justificáveis;
- Cancelamento de pedidos;
- Não possui demanda suficiente;
- Volatilidade de ganhos.

Parceiro:

Nome: Agnaldo
Idade: 31 anos
Gênero: Masculino
Ocupação: Parte do time NPS Churn

Bio:

Agnaldo é um homem casado, mas sem filhos, trabalha dentro da Rappi há apenas 6 meses e pensa bastante sobre seu futuro. Tem uma vida confortável e valoriza o tempo com a família. Possui Ensino Superior Completo em Administração, é um homem com bastante bagagem de conhecimento e capaz

de operar em vários setores do mercado. Tem um conhecimento prévio em algumas plataformas como Word, Excel, por sempre estar utilizando estas ferramentas em seu trabalho.

Principais Dores:

- Não há organização na base de dados;
- Dificuldade em saber quando um entregador dá “churn”

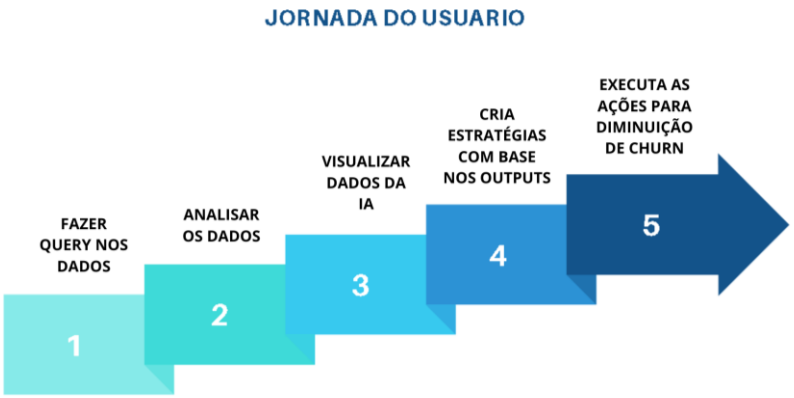
Necessidade na solução:

Um modelo de predição que mostre se um RappiTendero tem a possibilidade ou não de dar “churn” (ficar inativo por 21 dias na plataforma), já que há uma dificuldade de controlar o número de RT’s que “churneiam” ao longo do tempo.

Sua necessidade também está em entender os principais pontos e possíveis motivos que levam um RT a desistir da plataforma, ele tem uma grande base de dados, mas é desorganizada e não sabe como extrair informações dela. Gostaria de um modelo simples e objetivo, que seja bem explicativo em relação a como os dados estão sendo tratados para geração da saída final.

4.1.9. Jornadas do Usuário

A Jornada do Usuário é um modo de representação gráfica do relacionamento do cliente com um produto ou serviço de alguma empresa. Abaixo há o modelo proposto do funcionamento da solução final.



4.2. Compreensão dos Dados

Formato dos dados:

Todos os dados foram fornecidos no formato CSV, ao total temos 11 tabelas;

Tabelas fornecidas:

ABAIXO, SERIA POSSÍVEL VISUALIZAR OS DADOS FORNECIDOS PELO PARCEIRO PARA A CONSTRUÇÃO DO MODELO PREDITIVO. NO ENTANTO, EM RESPOSTA AO PEDIDO DO PARCEIRO PARA PROTEGER DADOS SENSÍVEIS, OS DADOS FORAM OCULTADOS.

4.3. Preparação dos Dados

Essa etapa consiste na separação e tratamento dos dados determinados para as “features” da Inteligência Artificial, a seleção foi realizada através da análise das principais causas para o churn dos RT’s. Logo abaixo serão apresentadas as etapas realizadas para definir os dados e os atributos descritivos dos dados a serem utilizados:

Churned:

Essa feature guarda informações de um entregador que já deu algum churn da plataforma da Rappi. Foi selecionada para entender a quantidade de RT’s que já saíram da plataforma ou não e analisar com outras features para relacionar um possível motivo. Não houve uma hipótese específica para seleção dessa feature, em virtude que ela é de grande importância por se relacionar diretamente com a saída final da solução.

O tratamento foi realizado, inicialmente, com a limpeza dos “NaN” (vazio) com o método “replace()” para substituir os valores “NaN” por 0. Essa substituição ocorreu para o entendimento da máquina durante o processamento de dados, visto que ela apenas interpreta números.

Após esse processo, os “ID” repetidos foram removidos com o “drop_duplicates()”, já que era necessário saber, apenas, em uma única linha quem está ou não na Rappi ainda.

Por último foi criada uma nova coluna “churned” dentro da tabela churn, com a finalidade de saber quem daquela tabela já está fora da plataforma.

Is Active:

Feature que apresenta se o RT, atualmente, está na plataforma ou não, pelo mesmo motivo da feature acima, não temos uma hipótese específica para essa.

Seu tratamento foi realizado, inicialmente, com o carregamento da tabela de info gerais, após isso a coluna “Transporte” foi renomeada para “Transport_Media_Type” (padrão da tabela de churn).

Foi definido que a tabela de info gerais teriam linhas de carro, moto ou bicicleta, para depois gerar o hot encoded para os tipos de transporte, com finalidade de apresentar através uma representação binária com novas colunas representando o transporte de cada entregador.

Um merge entre a tabela de churn e info gerais formando um novo data frame, com entregadores das informações de entregadores de info gerais que deram churn.

Os dados da Tabela de Info Gerais que não estão na tabela de Churn, definem seus valores para 'Churned' como 0 (Nunca deram Churn no Rappi), esse procedimento é para haver o processamento de dados nulos.

Tratamos os dados de 'Is Active' e 'Churned' que estavam como True para 1 (Padrão de dados que a IA entende).

Renomeamos a Coluna de 'ID' para 'STOREKEEPER_ID' (Padrão das outras tabelas).

Removemos todos IDs Duplicados da Tabela e resetamos seu Index.

Passamos o Data Frame para outra variável e limpamos as variáveis que não utilizamos mais (Para economizar memória).

Minutes Punishment e Count Punishments:

Será a feature que mostrará quantos minutos o RT levou de punição em toda sua trajetória no Rappi. A hipótese responsável por gerar essa feature foi que relacionar se o tempo de punição de um entregador tem a ver com o seu índice de churn.

A tabela foi tratada de modo a ser uma nova tabela com o ID do funcionário e a soma (sum()) dos tempos de punição, para saber o tempo total de punição de cada funcionário e quantas punições esse funcionário levou.

Foi feito um merge na tabela df principal (união das features) com o tempo e quantidade de punições de cada entregador, após utilizar o método rename para deixar o nome das colunas mais intuitivos. E houve a limpeza da tabela usando o “fillna()” para garantir que os NaN fossem substituídos por 0.

Age:

Feature com idade do RT, foi elaborada com base no entendimento se a idade do RT, interfere de alguma forma no índice de churn dele, talvez altere o tempo de espera ou ganhos.

Primeiro, é feito o carregamento da tabela de info gerais (tabela onde está a data de nascimento dos RT’s cadastrados), dividindo os dados em ano, mês e dia usando o método “split()”, com o intuito de utilizarmos apenas o ano de nascimento

Utilizando o ano de nascimento do entregador é feita uma subtração do ano atual para calcular a idade de cada entregador e após isso o método “drop()” remove os dados que não serão utilizados pela IA e por último é feito um merge com a tabela principal, com a feature “AGE” adicionada.

Distance to User:

Será a feacture que irá conter a distância média que um entregador percorre por pedido, em quilômetros. A hipótese para escolha dessa feacture foi baseada na distância média percorrida por um entregador, se essa distância interfere diretamente no churn, por fatores como aumento no gasto de combustível ou no trânsito.

Para trabalhar nessa feacture, foram usadas apenas as colunas de ID e deliveries count, e foi feita a soma do número de pedidos (deliveries count) por entregador através do agrupamento e depois a soma do número de pedidos usando o método sum().

Além disso, foi feita o somatório das distâncias percorridas por esses entregadores, para ter a divisão das da distância percorrida para cada entrega, para isso foi utilizada a divisão de dados de duas colunas para resultar em uma coluna nova “DISTANCE_MEDIAN”.

Índice de Churn (Churns Counts):

A hipótese elaborada pelo grupo sobre essa feature foi a seguinte:

Se um entregador dá churn com recorrência, ele pode tender a dar mais churn no futuro, ou ainda, se ele nunca deu churn, há alguma possibilidade que ele dê churn em breve. É uma feature de alta importância, pois mostra a regularidade com que o entregador saiu da plataforma e se houve retorno.

É uma coluna dentro da tabela principal que mostra a quantidade de churns por ID.

Frete Médio por pedido:

Feacture que apresenta o valor em dólares que o RT faz por pedido. A hipótese que baseou essa feature foi de que se o entregador não recebe ganhos suficientes para realizar uma entrega, ele pode tender a dar churn.

É uma coluna de ganho dos funcionários e o número de pedidos entregues por ele, chamada de “FRETE MÉDIO", ela foi selecionada da tabela info gerais e adicionada ao data frame principal.

Frete Médio por Distância:

Feacture que apresenta o valor médio que o RT recebe por distância percorrida. A hipótese que baseou essa feacture foi de que se o entregador não recebe ganhos suficientes para realizar uma entrega mais longa, ele pode tender a dar churn.

Primeiro, foi feito o carregamento da tabela de info gerais utilizando o “ID” e o “frete médio”, e feita a alteração do nome das colunas em questão.

Após isso, foi realizado o merge das tabelas em questão com a nova coluna “SHIPPING_MEDIAN” junto com “DELIVERIES_COUNT” e “DISTANCE_MEDIAN” que são os dados utilizados para gerar o novo data frame, e os valores dados ao “SHIPPING_MEDIAN”.

Foi separado a nova coluna “SHIPPING_PER_DISTANCE” que foi o resultado da coluna “SHIPPING_MEDIAN” dividido pela “DISTANCE_MEDIAN” resultando na nova coluna com o de arrecadação por entrega. Essa coluna traz a informação da renda do entregador pelo número de entregas realizadas por ele.z

Por último, foi feito um merge dessa nova feature com a tabela principal.

Taxa de cancelamento:

Feature que evidencia a proporção de entregas canceladas pelo próprio entregador em relação ao total de pedidos entregues ou cancelados pelo próprio cliente, em uma escala de 0 a 1. A hipótese que baseou essa feature foi para analisar se a taxa de cancelamentos de pedidos por entregador interfere na sua saída, visto que essa situação traz diversos transtornos para o RT.

Houve o carregamento da tabela de pedidos feitos e cancelados e criada uma nova coluna “CANCEL_RATE”, em que há a somatória dos pedidos cancelados por clientes e pelos próprios entregadores, dividido pelo total de pedidos feitos somados aos pedidos cancelados.

Ao final, foi feito um merge nessa nova coluna para o data frame principal.

Transporte utilizado:

Feature que mostra o tipo de transporte utilizado pelo RT, motocicleta, bicicleta ou carros. A hipótese que embasou essa feature foi que o risco ou gastos relacionados a esses modais de transporte podem ser importantes para o aumento no índice de churn.

O método utilizado para essa análise foi o hot encoded, em que foi dividida modalidade de transporte em 3 diferentes colunas com o tipo de transporte utilizado por cada RT e era adicionado 0 ou 1 a depender de qual transporte ele utilizava.

Após esse tratamento, essa feature foi adicionada ao data frame principal.

4.4. Modelagem

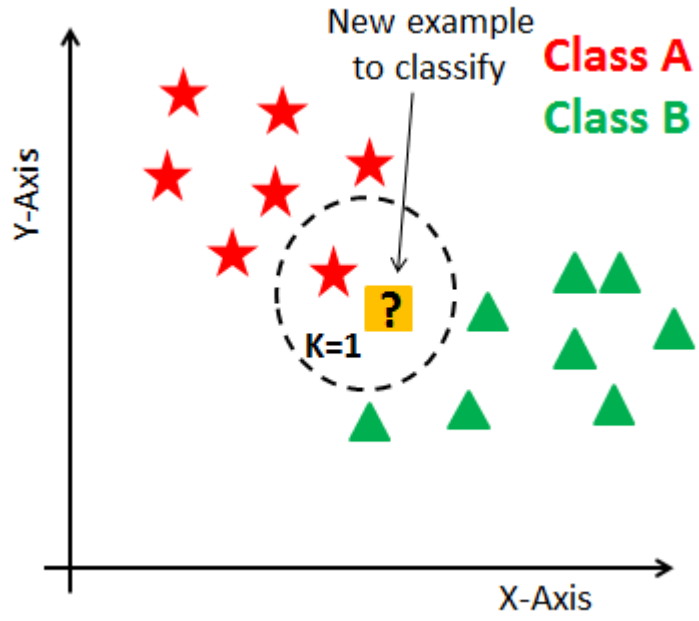
A métrica utilizada pelo grupo para avaliar os modelos escolhidos foi a de revocação, que tentará predizer o maior número de casos de Churns Verdadeiros, podendo, como consequência, encontrar mais casos de Falsos Churns. Esse modelo de avaliação foi o que mais se adequou à solução desejada pelo cliente.

Outra funcionalidade utilizada pelo grupo foram os hiperparâmetros, cuja finalidade é melhorar o desempenho do modelo selecionando valores para parâmetros já selecionados

O grupo selecionou alguns dos principais modelos para testar o novo data frame com as features selecionadas. Abaixo estão descritos os modelos utilizados para treinamento e teste do grupo:

1) K NEAREST NEIGHBOR (KNN):

É um algoritmo de aprendizado supervisionado — onde o conjunto de dados é rotulado — utilizado para resolver problemas de classificação e regressão, no nosso caso, foi utilizado para encontrar uma determinada classe dentro de um universo limitado de possibilidades — churn ou não churn — .



A figura acima ilustra o funcionamento do modelo KNN, dentro do modelo é definido um valor para uma variável K, que representa a quantidade de vizinhos selecionados pelo algoritmo e o modelo realiza sua predição baseado nessa proximidade.

Os passos utilizados pelo algoritmo são:

- Cálculo da distância entre os vizinhos;
- Encontrar os vizinhos mais próximos;
- Votar a label para o ponto em que será feita previsão.

```
from sklearn.neighbors import KNeighborsClassifier

# Instaciação do obj Algoritmo
modelo_knn = KNeighborsClassifier(n_neighbors=11)
# Treino # x = Features, y = Label/Target
modelo_knn.fit(x_train, y_train.squeeze()) # squeeze() -> df para series]

# Teste de Acuracidade (accuracy)
print('Acuracidade (treino): ', modelo_knn.score( x_train, y_train ))
print('Acuracidade (teste): ', modelo_knn.score( x_test, y_test ))
```

```
Acuracidade (treino):  0.8607192733650172
Acuracidade (teste):  0.8335859351318582
```

```
[74] y_pred = modelo_knn.predict(x_test)
     print('Revocação: ', recall_score(y_test, y_pred))
```

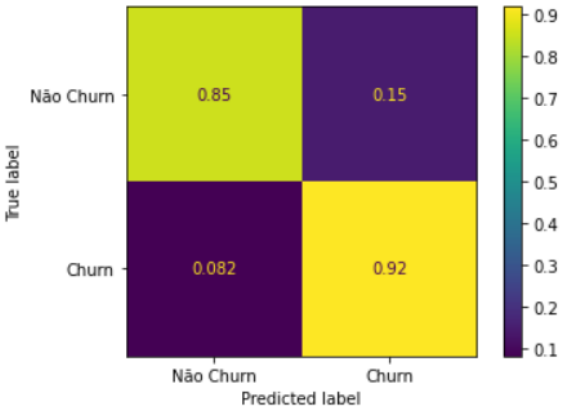
```
Revocação:  0.864516129032258
```

O modelo obteve resultados bem satisfatórios, porém outros modelos obtiveram resultados melhores em todos os requisitos solicitados. Portanto, o grupo optou por não utilizar hiperparâmetros nesse modelo, visto que não conseguiríamos um resultado satisfatório.

2) EXTRA TREES CLASSIFIER:

Cria muitas árvores de decisão de maneira aleatória, para então através da combinação dos resultados de cada árvore encontrar a resposta final. Seu principal diferencial está no fato deste processo ser extremamente aleatório, contribuindo assim para modelos mais generalizáveis.

A vantagem desse modelo é a diminuição de vieses, visto que suas análises são extremamente aleatórias, todavia sua desvantagem está relacionada com o uso de dados que muitas vezes não influenciam muito no seu target, justamente pela aleatoriedade do modelo.




```
from sklearn.ensemble import ExtraTreesClassifier

modelo_extrees = ExtraTreesClassifier()
modelo_extrees.fit( x_train, y_train.squeeze() )

print('Acertividade do treino: ', modelo_extrees.score(x_train, y_train))
print('Acertividade do teste: ', modelo_extrees.score(x_test, y_test.squeeze()))
```

```
Acertividade do treino:  0.9996048543374003
Acertividade do teste:  0.8785562724635171
```

```
[ ] y_pred = modelo_extrees.predict(x_test)

    print('Revocação: ', recall_score(y_test, y_pred))
```

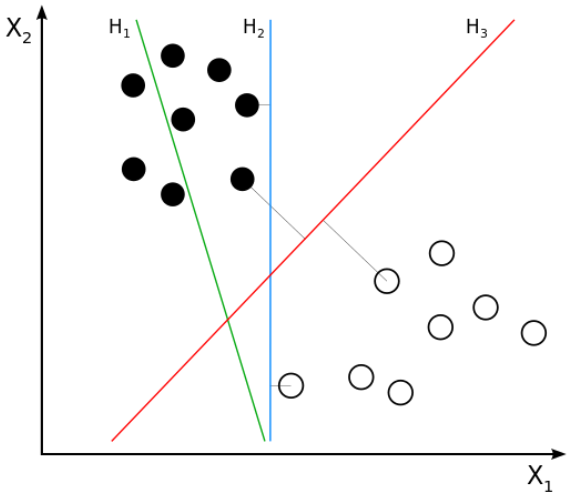
```
Revocação:  0.8989537925021795
```

Os resultados obtidos foram ótimos, mas há uma diferença considerável nos dados do teste e treino, mesmo que ambas estejam altas. Porém, esse modelo não foi escolhido pelo melhor desempenho de outros conforme o PyCaret, mas suas predições tem ótimos resultados.

Motivos para colocar hiperparâmetros: reduzir a diferença entre a acurácia de treino e teste nas predições, visto que ela estava superior a 10% e a IA estava viciada em relação aos dados relacionados ao treino, por isso seu desempenho nos testes era bem inferior.

3) SUPPORT VECTOR MACHINES (SVM):

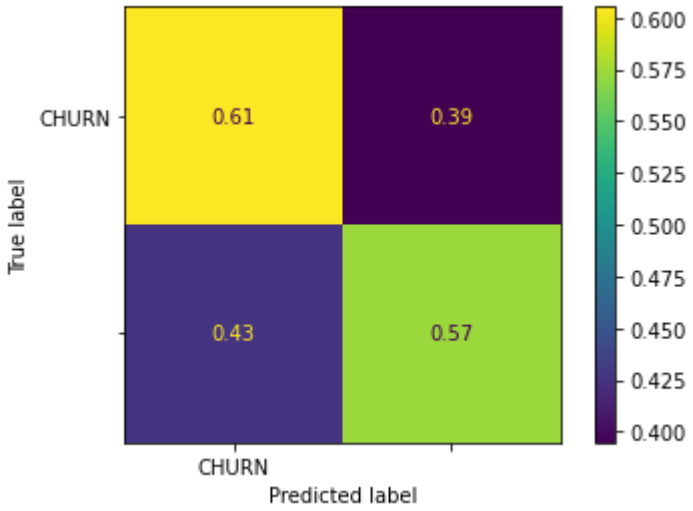
É um algoritmo que busca uma linha de separação entre duas classes distintas, analisando os dois pontos, um de cada grupo, mais próximos da outra classe. Isto é, o SVM escolhe a reta também chamada de hiperplano em maiores dimensões entre dois grupos que se distancia mais de cada um.



Após descoberta essa reta, o programa conseguirá predizer a qual classe pertence um novo dado ao checar de qual lado da reta ele está.

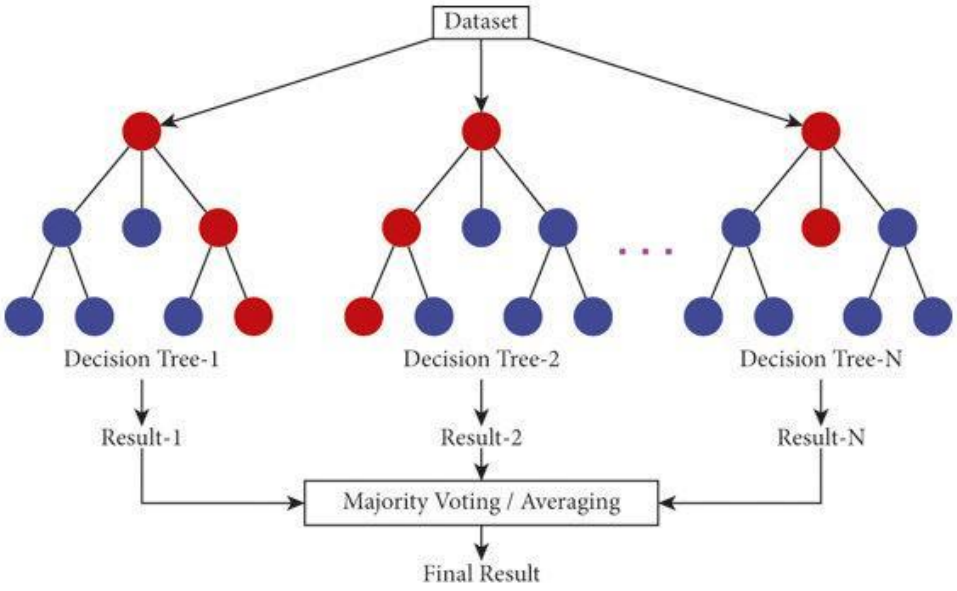
```
[76] from sklearn.svm import SVC

     modelo_svc = SVC(max_iter = 2000)
     modelo_svc.fit(x_train, y_train.squeeze())

     print('Acertividade do treino: ', modelo_svc.score(x_train, y_train))
     print('Acertividade do teste: ', modelo_svc.score(x_test, y_test.squeeze()))

     /usr/local/lib/python3.7/dist-packages/sklearn/svm/_base.py:289: ConvergenceWar
       ConvergenceWarning,
     Acertividade do treino:  0.5368676316159834
     Acertividade do teste:  0.5407699302819036

     y_pred = modelo_svc.predict(x_test)
     print('Revocação: ', recall_score(y_test, y_pred))

     Revocação:  0.1050566695727986
```

Esse modelo obteve uma baixa acurácia no momento de treino e teste, então não confiança nos resultados apresentados por suas predições. Além disso, não cumpre a métrica de revocação, que já era um pré-requisito para a Rappi. O grupo optou por não utilizar os hiperparâmetros nesse modelo, já que seus resultados foram extremamente negativos.



4) Random Forest:

Random forest cria múltiplas “árvores de decisão” aleatórias para o tratamento dos dados, de forma que ao final do processo, o dado seja classificado por meio de uma votação.

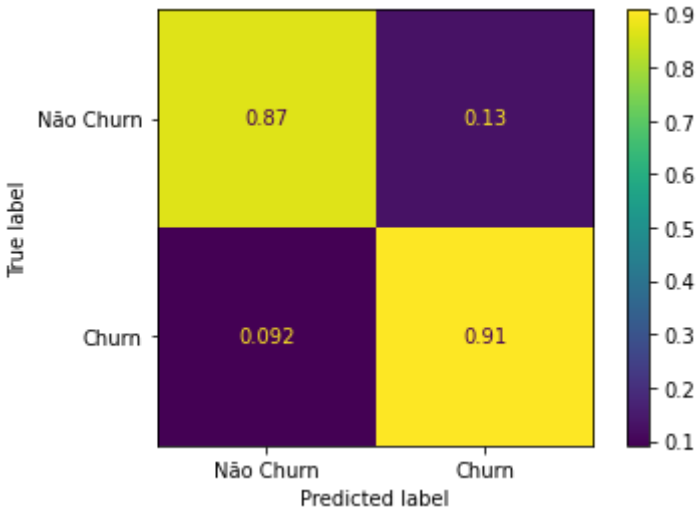



```
[79] from sklearn.ensemble import RandomForestClassifier

     modelo_rf = RandomForestClassifier()
     modelo_rf.fit( x_train, y_train.squeeze())

     print('Acertividade do treino: ', modelo_rf.score(x_train, y_train ))
     print('Acertividade do teste: ', modelo_rf.score(x_test, y_test.squeeze()))
```

```
Acertividade do treino:  0.9995940284288359
Acertividade do teste:  0.8748971549820291
```

```
y_pred = modelo_rf.predict(x_test)
print('Revocação: ', recall_score(y_test, y_pred))
```

```
Revocação:  0.8912380122057542
```



O modelo Random Forest também um obteve um resultado muito satisfatório, associado a uma excelente taxa de acerto em relação ao churn, como mostrado nos dados acima, porém seu erro está em 9,2% em relação aos casos de verdadeiros negativos (avisa que não será churn, mas ele dá churn), essa situação traz uma eficiência menor em relação ao objetivo do cliente, que é saber com a maior precisão possível quem dará churn para elaborar estratégias que reduzem a evasão dos RT's.

5) Regressão logística:

Na regressão logística, a probabilidade de ocorrência de um evento pode ser estimada diretamente. No caso da variável dependente Y assumir apenas dois possíveis estados (1 ou 0) e haver um conjunto de p variáveis independentes X1 , X2 , ... , Xp , o modelo de regressão logística pode ser escrito da seguinte forma:

$$P(Y=1)=\frac{1}{1+e^{-g(x)}}$$

g(x)= B 0 + B1X1 + · · · + Bp Xp

Os coeficientes B0 , B1 , ... , Bp **são estimados a partir do conjunto dados, pelo método da máxima verossimilhança,** em que encontra uma combinação de coeficientes que **maximiza a probabilidade** da amostra ter sido observada. Considerando uma certa combinação de coeficientes B 0 , B1 , ... , Bp e variando os valores de X. Observa-se que a **curva logística tem um comportamento probabilístico no formato da letra S**, o que é uma característica da regressão logística.

a) Quando

$$g(x) \to +\infty \quad \text{, então} \quad P(Y=1) \to 1$$

b) Quando

$$g(x) \to -\infty \quad \text{, então} \quad P(Y=1) \to 0$$

```
from sklearn.linear_model import LogisticRegression

modelo_lr = LogisticRegression(solver='lbfgs', max_iter=500)
modelo_lr.fit(x_train, y_train.squeeze())

print('Acertividade treino: ', modelo_lr.score(x_train, y_train))
print('Acertividade teste: ', modelo_lr.score(x_test, y_test.squeeze()))
```

```
Acertividade treino:  0.821469941864871
Acertividade teste:  0.8210280171480535
```

```
[65] y_pred = modelo_lr.predict(x_test)
     print( 'Revocação: ', recall_score( y_test, y_pred ))
```

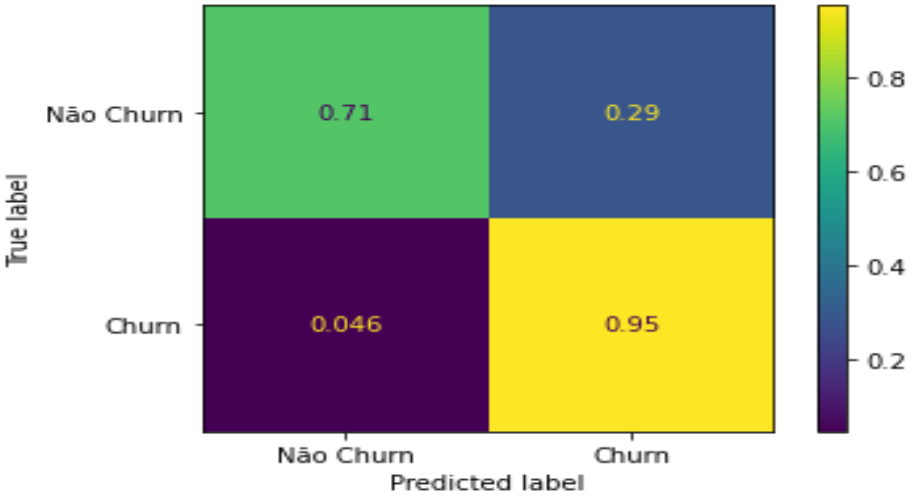
```
Revocação:  0.9570619006102877
```

Modelo com resultado foi muito satisfatório, foi um dos modelos escolhidos para colocação dos hiperparâmetros, pois sua resposta supriu as requisições de revocação e acuracidade do cliente.

Melhores Hiperparâmetros:

Os hiperparâmetros são parâmetros ajustáveis que permitem controlar o processo de treinamento do modelo, ou seja, aumentar a eficiência de requisitos, como acurácia, revocação, precisão, etc. Abaixo, há a seleção dos principais parâmetros de acordo com o algoritmo PyCaret.

boosting_type	gbdt
class_weight	None
colsample_bytree	1.0
importance_type	split
learning_rate	0.1
max_depth	-1
min_child_samples	20
min_child_weight	0.001
min_split_gain	0.0
n_estimators	100
n_jobs	-1
num_leaves	31
objective	None
random_state	8259
reg_alpha	0.0
reg_lambda	0.0
silent	warn
subsample	1.0
subsample_for_bin	200000
subsample_freq	0



Link do Google Colab contendo os modelos preditivos:

https://colab.research.google.com/drive/1STUvzITl4hOoeZHQM7obDE9K6vkkmeh0?usp=sharing

4.5. Avaliação

O modelo escolhido pelo grupo foi o de **Light Gradient Boosting (LGB)**, dado as necessidades apresentadas pelo cliente. Esse modelo foi escolhido pensando na menor incidência de falsos negativos (é dito que não haverá churn, mas o RT dá churn), esse modelo. É um modelo que traz uma economia de recursos para a Rappi, visto que é apresentado aqueles entregadores que darão churn com uma altíssima precisão, com isso a Rappi pode utilizar de alguma estratégia para tentar manter os entregadores ativos, talvez algum incentivo financeiro ou algum diálogo para entender o motivo da saída.

Ele foi escolhido através da análise de equilíbrio entre a Revocação e Acuracidade e o que apresentou o melhor balanceamento foi o modelo LGD, assim como mostrado na imagem abaixo, ou seja, apresentar um equilíbrio entre a performance geral do modelo dentre todas as classificações (acurácia) e a performance do modelo em relação ao acerto dos positivos (churns).

Foi possível perceber que há uma baixa taxa de erro do algoritmo em relação às principais métricas utilizadas para análise (revocação e acurácia). Já que o erro está relacionado a diferença total de 100% subtraído da taxa de acerto do modelo sendo assim:

Erro de Acurácia: 1 - 0.8536 = 0.1464

Erro de Revocação: 1 - 0.96 = 0.04

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lightgbm	Light Gradient Boosting Machine	0.8536	0.8752	0.9523	0.8727	0.9108	0.5081	0.5235	0.705
gbc	Gradient Boosting Classifier	0.8513	0.8643	0.9566	0.8674	0.9098	0.4913	0.5111	10.611
rf	Random Forest Classifier	0.8511	0.8638	0.9507	0.8712	0.9092	0.4995	0.5146	10.847
et	Extra Trees Classifier	0.8450	0.8495	0.9512	0.8647	0.9059	0.4719	0.4898	6.958
ada	Ada Boost Classifier	0.8439	0.8519	0.9522	0.8629	0.9054	0.4654	0.4845	2.482
lda	Linear Discriminant Analysis	0.8192	0.7782	0.9736	0.8267	0.8941	0.3016	0.3579	0.165
ridge	Ridge Classifier	0.8162	0.0000	0.9817	0.8196	0.8934	0.2647	0.3375	0.048
nb	Naive Bayes	0.8156	0.7723	0.9501	0.8368	0.8899	0.3362	0.3638	0.054
knn	K Neighbors Classifier	0.8153	0.7358	0.9385	0.8436	0.8885	0.3587	0.3769	0.666
lr	Logistic Regression	0.8097	0.6763	0.9570	0.8274	0.8875	0.2908	0.3281	0.780
svm	SVM - Linear Kernel	0.7981	0.0000	0.9363	0.8301	0.8788	0.2703	0.3042	4.021
dummy	Dummy Classifier	0.7844	0.5000	1.0000	0.7844	0.8792	0.0000	0.0000	0.043
dt	Decision Tree Classifier	0.7806	0.6834	0.8543	0.8644	0.8593	0.3613	0.3614	0.572
qda	Quadratic Discriminant Analysis	0.5242	0.4993	0.5354	0.7816	0.5766	0.0267	0.0290	0.075

Assertividade teste: 0.8536

Revocação: 0.9593

Esse foi o resultado final obtido com a implementação dos hiperparâmetros já selecionados para o modelo. Essa saída foi obtida utilizando a ferramenta Pycaret que classifica em um ranking os melhores modelos através do teste de vários parâmetros para cada modelo.

4.6 Comparação de Modelos

Nessa seção, alguns modelos foram selecionados e foram realizados alguns testes com aplicação de hiperparâmetros selecionados pelos próprios integrantes do grupo e avaliar qual o impacto deles dentro das predições. Os hiperparâmetros tem a finalidade de melhorar as características do modelo e o método de teste utilizado pelo grupo foi o de diversas tentativas com valores diferentes com busca pelos melhores resultados.

Nome: Light Gradient Boosting

Melhor assertividade teste:0.8844

Melhor revocação: 0.9199

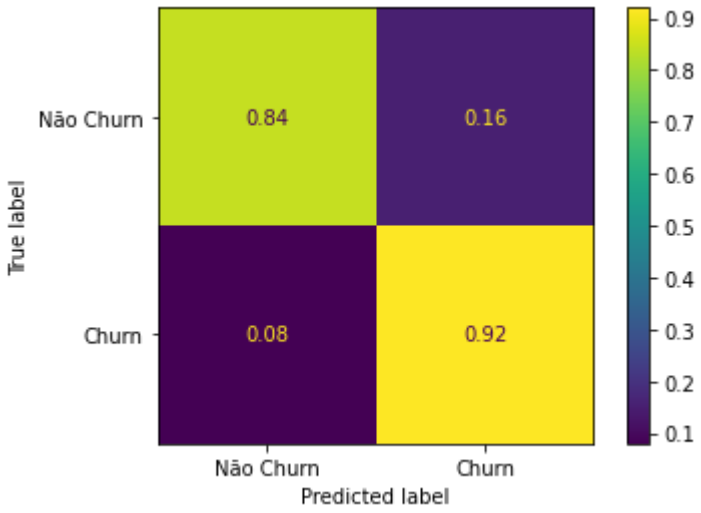
Motivos dos hiperparâmetros: Não havia um motivo em específico para o uso, a intenção era aprimorar a eficácia do modelo em questão.

Melhores features: AUTO_ACEITE, GORJETA, AGE, TRANSPORTE_bicycle, TRANSPORTE_car, TRANSPORTE_motorbike, CIDADE_Belo Horizonte, CIDADE_Grande São Paulo, CIDADE_Outros, CIDADE_Rio de Janeiro, CANCEL_RATE, MINUTES_PUNISHMENTS, NUM_PUNISHMENTS, DELIVERIES_COUNT, SHIPPING_PER_DISTANCE

Melhores hiperparâmetros:

- boosting_type: gbdt
- num_leaves: 30
- max_depth: 8
- random_state: 1
- learning_rate: 0.3
- n_estimators: 290
- importance_type: split
- n_jobs : 1

Matriz de confusão:



Nome: Random Forest

Melhor assertividade teste: 0.8243

Motivos dos hiperparâmetros: reduzir a diferença entre a acurácia de treino e teste nas predições, visto que ela estava superior a 10% e a IA estava viciada em relação aos dados relacionados ao treino, por isso seu desempenho nos testes era bem inferior.

Melhor revocação: 0.9140

Motivos dos hiperparâmetros: Tentar

Melhores features: AUTO_ACEITE, GORJETA, AGE, TRANSPORTE_bicycle, TRANSPORTE_car, TRANSPORTE_motorbike, CIDADE_Belo Horizonte, CIDADE_Grande São Paulo, CIDADE_Outros, CIDADE_Rio de Janeiro, CANCEL_RATE, MINUTES_PUNISHMENTS, NUM_PUNISHMENTS, DELIVERIES_COUNT, SHIPPING_PER_DISTANCE

Nome: Regressão Logística

Melhor assertividade teste: 0.8141

Motivo dos Hiperparâmetros: Não havia um motivo em específico para o uso, a intenção era aprimorar a eficácia do modelo em questão.

Melhor revocação:0.9432

Melhores features: AUTO_ACEITE, GORJETA, AGE, TRANSPORTE_bicycle, TRANSPORTE_car, TRANSPORTE_motorbike, CIDADE_Belo Horizonte, CIDADE_Grande São Paulo, CIDADE_Outros, CIDADE_Rio de Janeiro, CANCEL_RATE, MINUTES_PUNISHMENTS, NUM_PUNISHMENTS, DELIVERIES_COUNT, SHIPPING_PER_DISTANCE

Nome: Extra Trees Classifier

Motivos dos hiperparâmetros: reduzir a diferença entre a acurácia de treino e teste nas predições, visto que ela estava superior a 10% e a IA estava viciada em relação aos dados relacionados ao treino, por isso seu desempenho nos testes era bem inferior.

Melhor assertividade teste: 0.8229

Melhor revocação: 0.9225

Melhores features: AUTO_ACEITE, GORJETA, AGE, TRANSPORTE_bicycle, TRANSPORTE_car, TRANSPORTE_motorbike, CIDADE_Belo Horizonte, CIDADE_Grande São Paulo, CIDADE_Outros,

CIDADE_Rio de Janeiro, CANCEL_RATE, MINUTES_PUNISHMENTS, NUM_PUNISHMENTS, DELIVERIES_COUNT, SHIPPING_PER_DISTANCE

Melhores hiperparâmetros:

- 'n estimators': [10],
- 'bootstrap': [0, 1],
- 'oob_score': [0],

- 'random_state': [0],
- 'max_depth':[25],
- 'max_features': ['log2'],
- 'min_samples_split': [14]

Nome: Ridge Classifier

Melhor assertividade teste: 0.8149

Motivos dos hiperparâmetros: Esse modelo foi escolhido após o uso da ferramenta Picareta por trazer um resultado muito satisfatório.

Melhor revocação: 0.9801

Melhores features: AUTO_ACEITE, GORJETA, AGE, TRANSPORTE_bicycle, TRANSPORTE_car, TRANSPORTE_motorbike, CIDADE_Belo Horizonte, CIDADE_Grande São Paulo, CIDADE_Outros, CIDADE_Rio de Janeiro, CANCEL_RATE, MINUTES_PUNISHMENTS, NUM_PUNISHMENTS, DELIVERIES_COUNT, SHIPPING_PER_DISTANCE

5. Conclusões e Recomendações

A execução dos modelos preditivos será feita através do Google Colab que pode ser acessado através do link que está na seção “Anexos”. O documento está no formato de forms e é muito fácil de executar. Basta executar a primeira célula “Importação e conexão”, escolher qual modelo deseja usar e executar esta célula também. Após o carregamento das células, será mostrado um data frame com o ID do RT e probabilidade dele dar churn.

Solução Final:

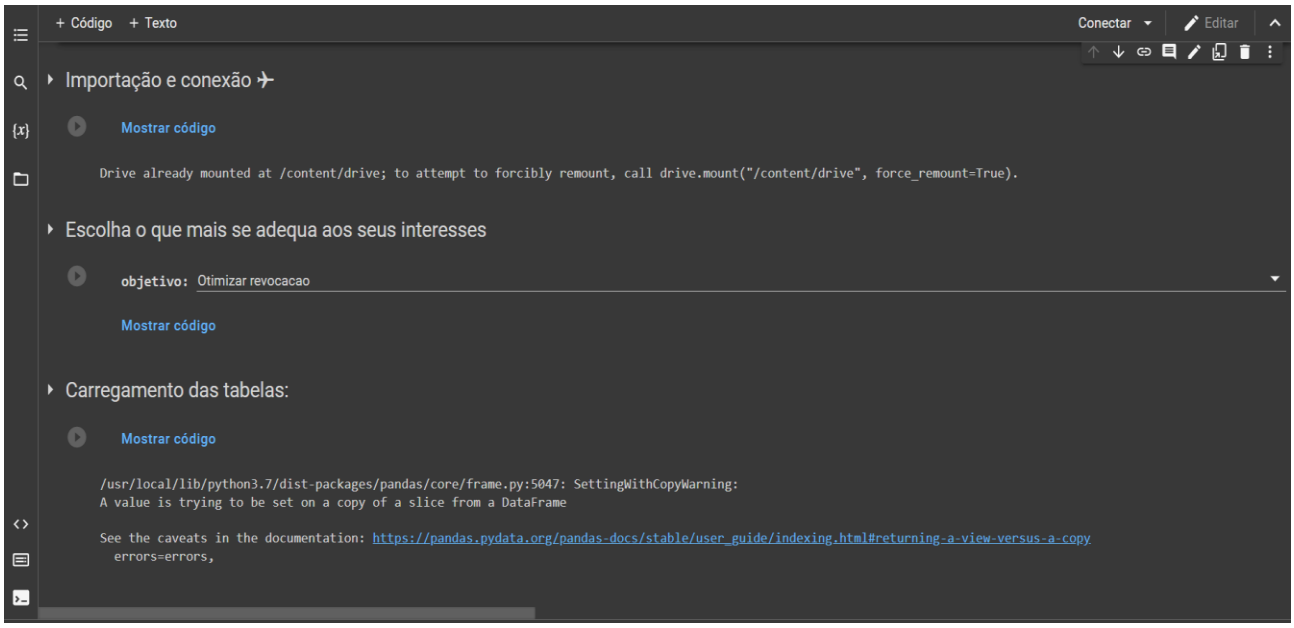
A entrega final está subdividida em dois colabs principais, a seguir está a explicação da entrega final disponibilizada pelo grupo:

O primeiro Colab está em formato de formulário, para evitar poluição visual no momento da utilização pelo time de operações, o código se torna oculto dentro da célula a ser executada. Os novos arquivos serão efetuados dentro desse Colab e nele haverá a saída com as previsões finais.

Sua composição é a seguinte:

- **Célula 1:** Importação e conexão, ou seja, todas as bibliotecas utilizadas no modelo estão importadas dentro desse campo, assim como a conexão com o Google Drive.

- **Célula 2:** Carregamento dos modelos escolhidos com base na característica e otimização gerada por ele. Os modelos utilizados foram os que houveram os melhores desempenhos em cada uma das três categorias: **Revocação, Acurácia, Precisão,** com isso foi escolhido um modelo para cada categoria citada, **Light Gradient Boosting (LGBM)(“maior certeza do resultado”) e Regressão Logística (“maior número de possíveis churn”).**

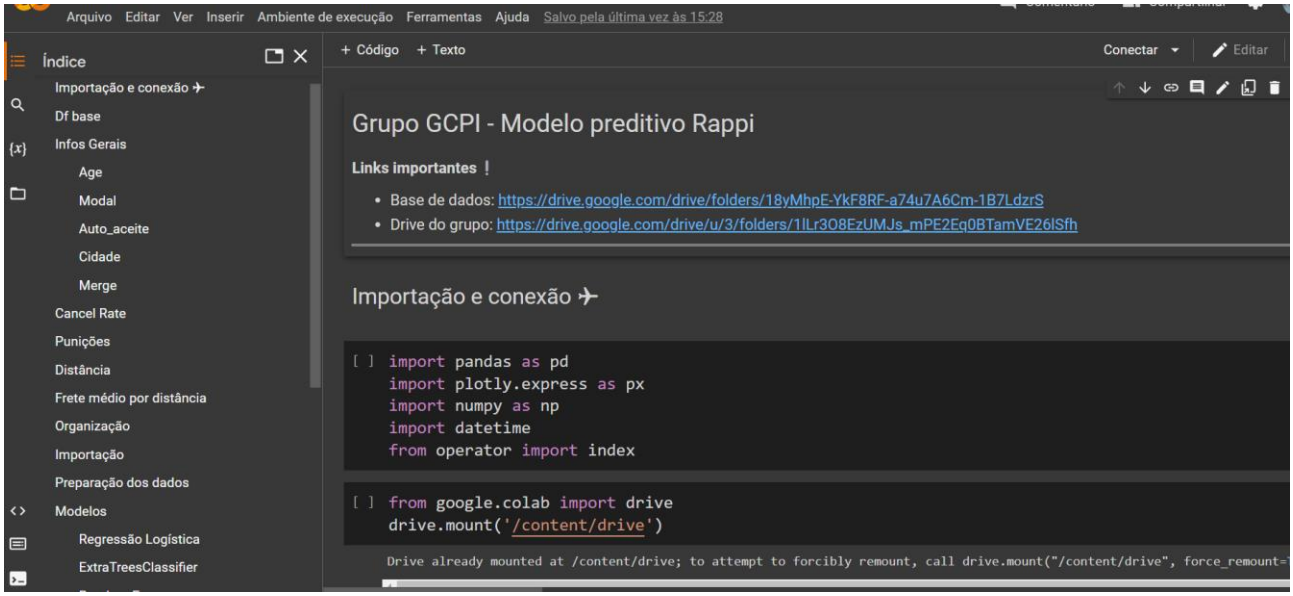


O resultado final da Sprint foi uma tabela com uma coluna de inatividade do RT (inativo ou não), outra coluna com a previsão apresentada pelo modelo, e uma classificação final que varia entre muito baixa e muito alta, em relação a probabilidade de cada entregador dar churn.

A tabela encontra-se salva dentro da pasta “Rappi” no Google Drive, juntamente com os Colabs de execução.

	A	B	C	D
1	ID	RT_INATIVO	PREVISTO_CHURN	PROBABILDADE_CH
2	1552668	Não	Sim	Muito Alta
3	1433164	Não	Sim	Muito Alta
4	412508	Não	Sim	Alta
5	1433033	Não	Sim	Alta
6	1433038	Não	Sim	Média
7	412474	Não	Sim	Muito Alta
8	1433078	Não	Sim	Alta
9	412331	Não	Sim	Alta
10	1433100	Não	Sim	Média
11	1433114	Não	Sim	Muito Alta
12	412177	Não	Sim	Alta
13	1433158	Não	Sim	Média
14	411980	Não	Sim	Alta
15	1433166	Não	Sim	Alta
16	412573	Não	Sim	Alta
17	411951	Não	Sim	Alta
18	1433179	Não	Sim	Média
19	1433181	Não	Sim	Alta
20	1433182	Não	Sim	Alta
21	411611	Não	Sim	Alta

O segundo Colab contém todo o tratamento de dados feito durante as Sprints, então há a elaboração das features, tratamento, preparação de dados, modelagem, hiperparâmetros, etc. Toda análise de negócio e de dados está dentro desse notebook.



Vale ressaltar a possibilidade de haver inconsistências nos dados e que os mesmos não devem ser usados sem a supervisão de um profissional. Dessa forma, a implementação deve ser feita com muita cautela e é importante ter algo em mente. Isso é, que os resultados preditos por uma IA nunca serão totalmente precisos, e por essa razão, não se é recomendado o seu uso sem supervisão, uma vez que pode afetar a vida de muitas pessoas e pode levar a resultados que podem ser considerados injustos.

6. Referências

seção 4.4 “**K-Nearest Neighbors** - KNN”

ALMEIDA, Alexandre *et al*. **O Algorítmo K-Nearest Neighbors (KNN) em Machine Learning**: o algoritmo do vizinho mais próximo. O Algoritmo do Vizinho mais Próximo. 2018. Disponível em: https://portaldatascience.com/o-algoritmo-k-nearest-neighbors-knn-em-machine-learning/. Acesso em: 5 out. 2022.

GUERRA, Bruno. **O que é preparação de dados e qual sua importância?**: para que serve a preparação de dados?. Para que serve a preparação de dados?. 2020. Disponível em: https://blog.in1.com.br/o-que-e-preparacao-de-dados-e-qual-sua-importancia. Acesso em: 12 set. 2022.

HOTZ, Nick. **What is CRISP DM?**: what are the 6 crisp-dm phases?. What are the 6 CRISP-DM Phases?. 2022. Disponível em: https://www.datascience-pm.com/crisp-dm-2/. Acesso em: 26 set. 2022.

FERREIRA JUNIOR, José Ribamar. **Hiperparâmetros**: o que é um hiperparâmetro?. O que é um hiperparâmetro?. 2021. Disponível em: https://www.linkedin.com/pulse/hiperpar%C3%A2metros-jose-r-f-junior/?originalSubdomain=pt. Acesso em: 20 set. 2022.

LightGBM (Light Gradient Boosting Machine) – Acervo Lima. Disponível em: <https://acervolima.com/lightgbm-light-gradient-boosting-machine/>. Acesso em: 6 out. 2022.

Anexos

Google Drive contendo os Colabs para a execução dos modelos e local onde bases de dados devem ser inseridas.

https://drive.google.com/drive/folders/1QAwfG64_h2sMujFuwDWvDGpFChBoHal9?usp=sharing