



MODELO PREDITIVO DE CHURN DE ENTREGADORES

EMPRESA RAPPI

Controle do Documento

Histórico de revisões

| Data | Autor | Versão | Resumo da atividade |
|------------|---|--------|---------------------------------------|
| 08/08/2022 | -Alysson Cordeiro; -Bruno Moitinho Leão; -Frederico Schur; -Israel Carvalho; -Luiz Carlos da Silva Júnior; -Stefano Tosi Butori. | 1.0 | Criação do documento |
| 09/08/2022 | Todos | 1.1 | Atualização dos tópicos 4.1.1 a 4.1.5 |
| 10/08/2022 | Todos | 1.2 | Atualização do tópico 4.2 |

Sumário

| | |
|--|----------|
| 1. Introdução | 5 |
| 2. Objetivos e Justificativa | 6 |
| 2.1. Objetivos | 6 |
| 2.2. Justificativa | 6 |
| 3. Metodologia | 7 |
| 3.1. CRISP-DM | 7 |
| 3.2. Ferramentas | 7 |
| 3.3. Principais técnicas empregadas | 7 |
| 4. Desenvolvimento e Resultados | 8 |
| 4.1. Compreensão do Problema | 8 |
| 4.1.1. Contexto da indústria | 8 |
| 4.1.2. Análise SWOT | 11 |
| 4.1.3. Planejamento Geral da Solução | 12 |
| 4.1.4. Value Proposition Canvas | 13 |
| 4.1.5. Matriz de Riscos | 15 |
| 4.1.6. Personas | 17 |
| 4.1.7. Jornada do Usuário | 19 |
| 4.2. Compreensão dos Dados | 20 |
| 4.2.1. Descrição Geral dos Dados | 20 |
| 4.2.2. Estatística Descritiva dos Dados | 27 |
| 4.2.3. Descrição da Predição Desejada ("Target") | 31 |
| 4.3. Preparação dos Dados | 32 |
| 4.3.1. Manipulação | 32 |
| 4.3.2. Agregação | 32 |
| 4.3.3. Remoção e Substituição | 33 |
| 4.3.4. Seleção das features | 33 |

| | |
|--|-----------|
| 4.4. Modelagem | 38 |
| 4.4.1. Avaliação e Seleção de Modelos (Algoritmos) | 39 |
| 4.4.2. Modelos e Configuração de Hiperparâmetros | 41 |
| 4.4.3. Resultados Obtidos | 41 |
| 4.5. Avaliação | 42 |
| 4.5.1. Análise dos Resultados | 42 |
| 4.6 Comparação de Modelos | 48 |
| 5. Conclusões e Recomendações | 49 |
| 6. Referências | 50 |
| Anexos | 51 |

1. Introdução

Apresente de forma sucinta o parceiro de negócio, seu porte, local, área de atuação e posicionamento no mercado. Maiores detalhes deverão ser descritos na seção 4

Descreva resumidamente o problema a ser resolvido (sem ainda mencionar a solução).

Caso utilize citações ao longo desse documento, consulte a norma ABNT NBR 10520. Sugerimos o uso do sistema autor-data para citações.

Rappi é uma startup unicórnio colombiana estabelecida na América Latina. O seu modelo de negócios é baseado em entregas por demandas. Criada em 2015 com o propósito de fazer compras de supermercados de maneira fácil e rápida, a Rappi é uma das maiores empresas do setor de delivery, tendo uma participação significativa no mercado. No momento o seu maior problema é a falta de entregadores na plataforma, devido à saída dos mesmos.

2. Objetivos e Justificativa

2.1. Objetivos

Descreva resumidamente os objetivos gerais e específicos do seu parceiro de negócios

2.2. Justificativa

Faça uma breve defesa de sua proposta de solução, escreva sobre seus potenciais, seus benefícios e como ela se diferencia.

3. Metodologia

Descreva as etapas metodológicas que foram utilizadas para o desenvolvimento, citando o referencial teórico. Você deve apenas enunciar os métodos, sem dizer ainda como ele foi aplicado e quais resultados obtidos.

3.1. CRISP-DM

Descreva brevemente a metodologia CRISP-DM e suas etapas de processo

3.2. Ferramentas

Descreva brevemente as ferramentas utilizadas e seus papéis (Google Collaboratory)

3.3. Principais técnicas empregadas

Descreva brevemente as principais técnicas empregadas, algoritmos e seus benefícios

4. Desenvolvimento e Resultados

4.1. Compreensão do Problema

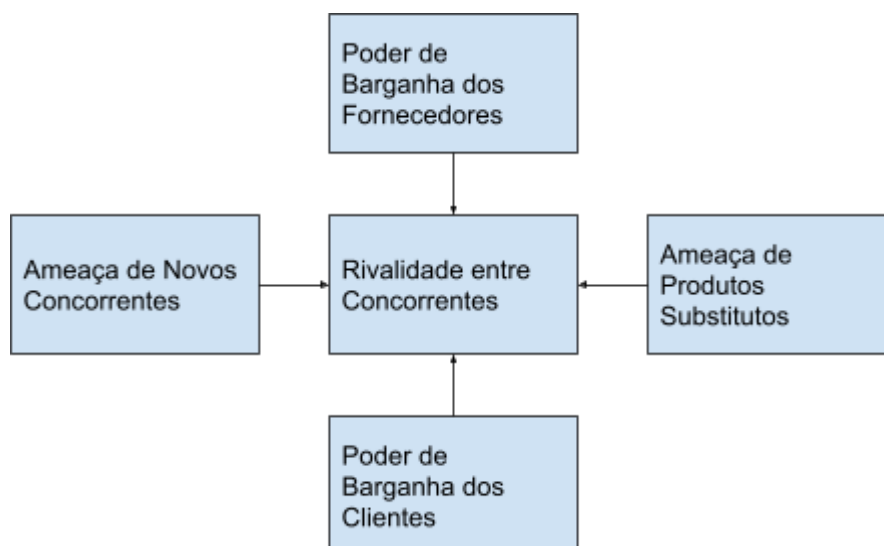
4.1.1. Contexto da indústria

O modelo de negócio da Rappi é do tipo plataforma multilateral, pois conecta estabelecimentos comerciais a entregadores e a clientes que precisam de serviços de entrega (delivery). A plataforma disponibiliza cardápios e menus das mais variadas instalações - restaurantes, lojas de todos os ramos, supermercados, entre outros.

Além do mais, a concorrência da companhia é bastante abrangente, porque, de certa forma, inclui desde toda e qualquer empresa de delivery (especialmente os aplicativos de entregas, como, por exemplo, IFood, Loggi, Glovo e Lalamove) e até os e-commerces gigantes, como a Amazon; esse de uma forma indireta.

Vale ressaltar, ademais, que o delivery é uma das principais tendências do mercado após o isolamento social da pandemia do COVID-19 em 2020. E ele tende apenas se fortalecer. No entanto, as entregas estão deixando de ser focadas apenas em comida para alcançar novos patamares, como medicamentos e outros produtos diversos, por exemplo. Além disso, esse tipo de mercado está optando por uma prática mais sustentável, como, por exemplo, ajudar o consumidor a identificar restaurantes que usam menos plásticos em seus produtos e tornar grande parte dos pedidos neutros em carbono (CO2) no país.

Utilizaremos a análise das 5 Forças de Porter para entender melhor a indústria de delivery por aplicativo.



Ameaça de Novos Concorrentes:

Durante a pandemia, 89% dos estabelecimentos passaram a utilizar o delivery como uma solução para disponibilizar seus produtos (grande crescimento do mercado). Entretanto, o alto custo operacional gerado pela operação no mercado de delivery acaba dificultando a instalação de novas empresas, além da manutenção de empresas já estabelecidas, como por exemplo o recente caso do Uber Eats (alto custo operacional).

Poder de Barganha dos Clientes:

Por mais que houve um aumento expressivo no setor de delivery impulsionado principalmente pela necessidade das empresas de se adequarem às restrições impostas ao SARS-CoV-2 (coronavírus) e, também, considerando o cenário macroeconômico dos últimos meses, é possível notar um aumento da inflação e, conseqüentemente, uma diminuição no grau de consumo em diversas áreas da economia, incluindo a de delivery. Logo, é razoável inferir que o poder de barganha dos consumidores nesse cenário é relativamente baixo, visto que diante de um cenário em que há uma alta de preços em toda a economia, os consumidores não conseguirão “barganhar” por preços menores.

Ameaça de Produtos Substitutos:

Não há muitos produtos substitutos para o delivery por aplicativo. A opção alternativa mais evidente seria a entrega efetivada diretamente em contato com o restaurante, eliminando a intermediação via aplicativo. Essa opção não oferece a comodidade de ver o cardápio de opções pelo aplicativo e apresenta o incômodo de ter que pesquisar um restaurante e conversar com um atendente por telefone.

Poder de Barganha dos Fornecedores:

Tendo em vista todo o cenário descrito nos tópicos anteriores, é possível concluir que o poder de barganha dos fornecedores é expressivo, uma vez que eles controlam em qual marketplace estarão disponíveis. Por outro lado, contratos de exclusividade podem restringir a capacidade de escolha dos fornecedores e reduzir o poder de barganha destes.

Rivalidade entre Concorrentes:

A rivalidade entre os concorrentes nesse setor é extremamente alta em consequência do constante esforço para garantirem uma operação de qualidade, além de um grande market share. É comum que empresas desse setor gastem seus recursos de caixa para crescer, e, conseqüentemente, acabam não dando lucro.

4.1.2. Análise SWOT

| | |
|--|--|
| <p style="text-align: center;">Strengths (Forças)</p> <ul style="list-style-type: none"> - Entrega turbo em 10 minutos - Alguns parceiros exclusivos - Parceria com players de outros setores (ex: HBOMax) - Presença substancial na América Latina - Atuação em outros setores, como o financeiro (RappiBank) | <p style="text-align: center;">Weakness (Fraquezas)</p> <ul style="list-style-type: none"> - Dependência de entregadores que não são CLT - A função de rastreo da entrega não é consistente - “Churn” elevado de entregadores |
| <p style="text-align: center;">Opportunities (Oportunidades)</p> <ul style="list-style-type: none"> - Saída recente de um player relevante, abrindo assim uma nova fatia do mercado - Aumento da demanda por delivery de diversos produtos | <p style="text-align: center;">Threats (Ameaças)</p> <ul style="list-style-type: none"> - Mercado bastante competitivo - Instabilidade de fatores externos que afetam os entregadores, como preço da gasolina, variações climáticas ou feriados |

4.1.3. Planejamento Geral da Solução

Dados disponíveis: A Rappi disponibilizou até o momento uma série de arquivos CSV, com o conteúdo detalhado de forma mais extensiva no tópico 4.2.1 deste documento. Os dados desses arquivos, conforme informações repassadas pelo cliente em entrevista, foram obtidos por meio do aplicativo de delivery.

Esses arquivos incluem informações diversas sobre os entregadores, como a taxa de aceitação dos pedidos, receita mensal, reclamações sobre pedidos, data de criação das contas que sofreram "churn", registro de suspensões e avisos, número de ordens, tempo que o entregador ficou "online", ordens devolvidas e dados gerais.

Solução proposta: Pretendemos desenvolver um modelo preditivo para classificar se um entregador irá permanecer ou sair ("churn") da plataforma Rappi. Como efeito secundário, podemos verificar a probabilidade que o modelo entende que aquele resultado se concretizará e os fatores ("features") que o modelo considera para a classificação geral.

Tipo de tarefa (regressão ou classificação): a predição de "churn" é essencialmente uma tarefa de *classificação binária* - o objetivo é informar se os entregadores irão: (i) permanecer; ou (ii) sair ("churn") da plataforma dentro de um período determinado.

Forma de utilização da solução proposta: a Rappi poderá usar o modelo para classificar se um entregador está propenso a sair da plataforma e, se desejar, dedicar mais esforços para manter esse entregador na plataforma.

Benefícios trazidos pela solução proposta: o modelo pode permitir uma visualização rápida dos entregadores mais inclinados a parar de usar a plataforma em breve, levar a um maior entendimento dos fatores que podem causar esse evento, facilitar o diálogo e reduzir a taxa de "churn" para um patamar adequado.

Critério de sucesso e métricas de avaliação: entendemos que o modelo deve apresentar certa probabilidade de acerto em suas predições. Para tanto, utilizamos como métricas de avaliação:

- A. acurácia ("accuracy"), a razão entre o número de predição corretas e o número total de predições, fornece um indicador de confiabilidade geral do modelo:

$$\frac{\text{Positivos Verdadeiros} + \text{Negativos Verdadeiros}}{\text{Positivos Verdadeiros} + \text{Negativos Verdadeiros} + \text{Positivos Falsos} + \text{Negativos Falsos}};$$
- B. precisão ("precision"), a confiabilidade do modelo quando ele aponta que um resultado é positivo:

$$\frac{\text{Positivos Verdadeiros}}{\text{Positivos Verdadeiros} + \text{Positivos Falsos}};$$
- C. revocação ("recall"), a confiabilidade do modelo em detectar *todos* os resultados positivos corretamente:

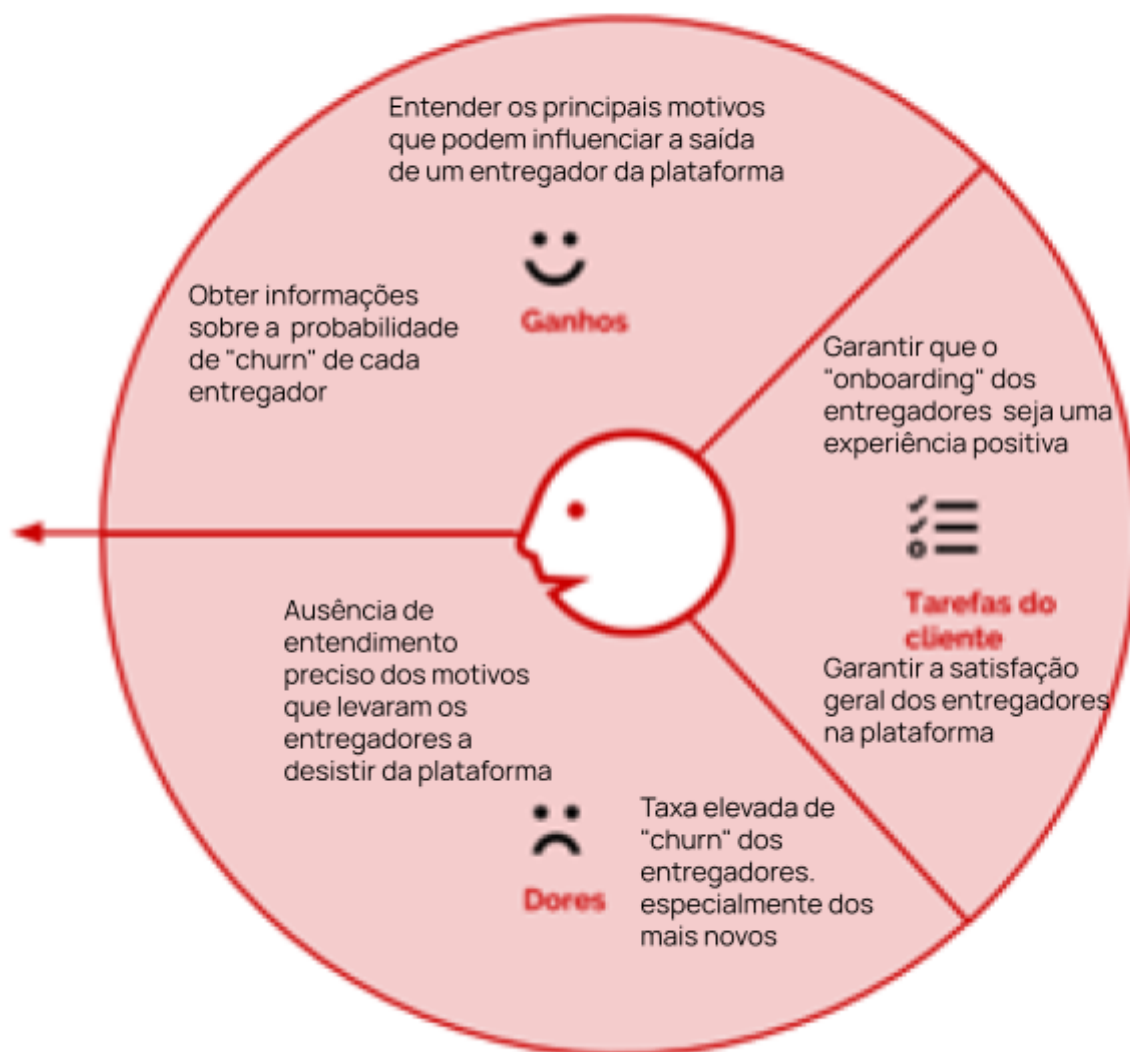
$$\frac{\text{Positivos Verdadeiros}}{\text{Positivos Verdadeiros} + \text{Negativos Falsos}};$$

4.1.4. Value Proposition Canvas

Proposta de Valor



Perfil do Cliente



4.1.5. Matriz de Riscos

| | | Ameaças | | | | |
|---|-----|-------------|-------|---|---|---|
| P r o b a b i l i d a d e | 90% | | | Bugs no código do modelo preditivo | | |
| | 70% | | | | | |
| | 50% | | | Imprecisão / inconsistência dos dados usados para análise | | |
| | 30% | | | Não ter dados relevantes para solucionar o problema | Modelo ter uma precisão ou acurácia baixa | |
| | 10% | | | | | Sistema exigir muita capacidade computacional |
| | | Muito baixo | Baixo | Moderado | Alto | Muito alto |
| | | Impacto | | | | |

| | | Oportunidades | | | | |
|---|-----|--|--|---|-------|-------------|
| P r o b a b i l i d a d e | 90% | | | | | |
| | 70% | Concorrência não possuir ferramenta que mapeie o churn | Sistema prevê corretamente a chance de saída dos RTs | Bom volume de dados para treinar o modelo | | |
| | 50% | | | | | |
| | 30% | | | | | |
| | 10% | | | | | |
| | | Muito alto | Alto | Moderado | Baixo | Muito baixo |
| | | Impacto | | | | |

4.1.6. Personas

José González



| | |
|--------------|------------------------------------|
| AGE | 29 |
| EDUCATION | Gestão da Tecnologia da Informação |
| STATUS | Solteiro |
| OCCUPATION | Diretor de Operações |
| LOCATION | São Paulo |
| TECH LITERTE | Alta |

“Estou acostumado com atendimento ao cliente, principalmente entender o comportamento do cliente em relação ao produto.

Personalidade

Comunicativo

Simpático

Biografia

Mora em São Paulo, na capital. Graduiu-se em Sistemas da Informação, terminou sua pós em Gestão de TI em 2020. Se especializou em customer experience. Devido a sua especialização, foi promovido recentemente em uma empresa de Delivery. É um homem comunicativo, empreendedor e está sempre antenado para as novas tendências e oportunidades de mercado. González atualmente está solteiro e gosta de viajar sempre que pode.

Necessidades

- Entender o motivo do elevado nível da saída de entregadores de aplicativo da empresa.
- Pretende utilizar um modelo preditivo para classificar se um entregador permanecerá ou sairá da companhia.

Frustração

- Baixo contato com entregadores de delivery.
- Possui dificuldade em analisar a probabilidade de “churn” dos entregadores.
- Não tem entendimento exato dos motivos que levaram os entregadores a desistir da plataforma.

Hobbies

Viajar; leitura; aprender novos idiomas; esportes.

Motivações

Liderança; conhecimento; Ter flexibilidade na carreira; Impactar os clientes com os resultados de seus projetos.

Cleyton Soares



| | |
|---------------|------------------------|
| AGE | 21 |
| EDUCATION | Ensino Médio completo |
| STATUS | Solteiro |
| OCCUPATION | Entregador de delivery |
| LOCATION | São Paulo |
| TECH LITERATE | Média |

“ Há 2 anos estou na empresa de delivery como entregador. E com isso desenvolvi uma boa comunicação e relação com os clientes.

Personalidade

Extrovertido Responsável

Biografia

Cleyton mora na Zona Leste de São Paulo. Terminou seu Ensino Médio em 2019. Entrou para Rappi para trabalhar como entregador após ser demitido durante a pandemia. É um jovem bem extrovertido e comunicativo. Gosta de bater-papo com os amigos.

Necessidades

- Maximizar os ganhos na plataforma
- Ter mais pedidos de entregas concluídos
- Contratação de seguros de vida da empresa de delivery

Frustração

- Insatisfação quando o cliente cancela o pedido
- Aplicativo da empresa mostra a rota do GPS diferente do local desejado
- Sua remuneração é desproporcional ao valor da inflação da gasolina

Hobbies

Esportes; Assistir séries; jogar games eletrônicos; ir à praia.

Motivações

Iniciativa; determinação e pontualidade em entregar os produtos pedidos pelos clientes.

4.1.7. Jornada do Usuário



Jose González

Cenário

Jose é um diretor de operações da Rappi e percebe que hoje a empresa tem uma grande evasão de entregadores ("RTs"). Assim, deseja criar uma ferramenta de IA que consiga prever as chances de um RT sair da plataforma

Metas

- Prever e antecipar a saída dos RTs
- Entender os motivos da saída dos RTs
- Criar ações e medidas para a retenção dos RTs

Extrair

- Levantar dados e informações necessárias dos RTs na base de dados da Rappi

"Que dados devo obter para alimentar o sistema adequadamente?"

1

Popular

- Popular o notebook de processamento de features preparado pelo grupo

"Vou inserir os dados em um notebook para processamento"

2

Carregar

- Carregar os dados no modelo, usando o notebook de modelagem preparado pelo grupo

"Agora vou carregar os dados no modelo"

3

Resultado

- Obter uma predição da possibilidade de saída dos RTs
- Compreensão dos motivos de saída dos RTs
- Ações para a permanência dos RTs

"Que interessante, esses entregadores tem maior probabilidade de sair da plataforma"

4

Oportunidades

- Auxiliar na tomada de decisão dos gerentes
- Ajudar os RTs a terem seus problemas resolvidos

Métricas

- Reduzir o "churn" de RTs para um nível aceitável
- Aumentar o Net Promoter Score ("NPS") dos RTs

4.2. Compreensão dos Dados

4.2.1. Descrição Geral dos Dados

Os dados disponibilizados pelo cliente consistem em diversos arquivos "Comma-separated values" ("CSV"), elencados na tabela abaixo. Segundo informações repassadas pela Rappi, todos os dados foram obtidos através do aplicativo. Também descrevemos na tabela o que cada arquivo representa, em geral, e o seu tamanho, com número de linhas e de colunas.

| Nome do CSV | Descrição Geral do Conteúdo | Número de Linhas | Número de Colunas |
|-----------------------------|--|------------------|-------------------|
| 2022-08-10 10_40am | A distância (em km) que o entregador percorreu para entregar cada pedido | 31.382.215 | 4 |
| attendance rate | Taxa de aceitação dos pedidos. Ex: tocou 10 vezes, aceitei 9, 90% | 653.167 | 2 |
| comp defects | Hoje, sempre que um usuário abre alguma reclamação sobre algum pedido incompleto, faltante, item errado. | 6.783.958 | 10 |
| criacao contas churn-002 | Data de criação das contas churn do período | 32.568.384 | 9 |
| earnings | Receita de cada entregador | 566.099 | 4 |
| Incidentes_Regras RT | Existem diversas regras para melhoria da qualidade da operação. Exemplo: A regra 92/93 remove/libera o RT do pedido caso ele não esteja em movimento ou em direção ao cliente. (Reforço que não temos informação sobre a localização do RT, apenas se esta | 2.405.601 | 9 |

| | | | |
|---------------------------|---|------------|----|
| | diminuindo o ETA ou não). | | |
| infos gerais | Informações gerais do entregador | 180.178 | 25 |
| Ordens Done e Cancel | ordens realizadas e ordens canceladas; Ordens podem ser canceladas por qualquer razão: pela loja, falta de produto, pelo RT (Pneu furado, roubo, acidente, problemas pessoais, etc). | 653.166 | 4 |
| Product return | Retorno de produto uma vez que a ordem foi cancelada. Ex: comprei itens de supermercado e por qualquer razão ordem foi cancelada; o RT precisa retornar este a loja (nem todas aceitam, ponto ruim) ou devolver em algum ponto de apoio Rappi; Até isso acontecer, ele fica com uma dívida no valor dos produtos; | 41.535 | 11 |
| supply | Tempo em Horas que o RT fica/ficou conectado no período; | 124.526 | 10 |
| tempo resolucao e modal 2 | Quanto tempo o entregador ficou esperando resolução para algum pedido aberto junto ao suporte Central Rappi | 32.568.384 | 9 |

Mais especificamente, podemos destacar o que cada coluna significa nos arquivos CSV, conforme pode ser visto na tabela abaixo:

| CSV | Nome da Coluna | Descrição | Tipo |
|--------------------|----------------|------------------|---------|
| 2022-08-10 10_40am | ORDER_ID | ID do pedido | int64 |
| 2022-08-10 10_40am | STOREKEEPER_ID | ID do entregador | float64 |

| | | | |
|--------------------------|---------------------|---|---------|
| | | | |
| 2022-08-10 10_40am | DISTANCE_TO_USER | Distância (em KM) do usuário que fez o pedido | float64 |
| 2022-08-10 10_40am | BUNDLE_ID | ID do "pacote" de pedidos | object |
| attendance rate | STOREKEEPER_ID | ID do entregador | float64 |
| attendance rate | ACCEPTANCE_RATE | Porcentagem de pedidos aceitos pelo entregador em relação ao total de pedidos recebidos | float64 |
| comp defects | STOREKEEPER_ID | ID do entregador | float64 |
| comp defects | WEEK | Segunda-feira da semana em referência (YYYY-MM-DD) | object |
| comp defects | CITY | Cidade | object |
| comp defects | LEVEL_ID | ID do nível do entregador | float64 |
| comp defects | LEVEL_NAME | Nome do nível do entregador | object |
| comp defects | ORDERS | Número de pedidos que aconteceu algum problema | int64 |
| comp defects | GMV_TOTAL | Total da transação (GMV = custo total pago pela Rappi para a loja) | float64 |
| comp defects | COMPENSATIONS | Valor pago (devolvido) para o usuário | float64 |
| comp defects | ID_COMPENSATIONS | ID do pedido que deu problema e foi compensado | float64 |
| comp defects | DEFECT_ORDER | ID do pedido que deu problema e foi compensado | float64 |
| criacao contas churn-002 | ID | ID do entregador | int64 |
| criacao contas churn-002 | FIRST_NAME | Nome do entregador | object |
| criacao contas churn-002 | GENDER | Gênero do entregador | object |
| criacao contas churn-002 | CITY | Cidade de atuação | object |
| criacao contas churn-002 | SK.CREATED_AT::DATE | Data de cadastro da conta | object |

| | | | |
|--------------------------|------------------------|--|---------|
| | | | |
| criacao contas churn-002 | TRANSPORT_MEDIA_TYPE | Modal de transporte | object |
| criacao contas churn-002 | CARTAO | Tem cartão pré-pago? | object |
| criacao contas churn-002 | LEVEL_NAME | Nível do entregador | object |
| criacao contas churn-002 | FECHA_ULT | Última data em que o entregador teve alguma interação com o aplicativo | object |
| earnings | STOREKEEPER_ID | ID do comerciante | int64 |
| earnings | MONTH | Primeiro dia do Mês, em formato ISO-8601 | object |
| earnings | EARNINGS | Receita do entregador | float64 |
| earnings | TIPS | Gorjetas do entregador | float64 |
| Incidentes_Regras RT | DATE | Data do Incidente (formato YY-MM-DD) | object |
| Incidentes_Regras RT | NAME | Nome do Incidente | object |
| Incidentes_Regras RT | INCIDENT_ID | ID do incidente | int64 |
| Incidentes_Regras RT | STOREKEEPER_ID | ID do entregador | int64 |
| Incidentes_Regras RT | PUNISHMENT_MINUTES | Minutos de suspensão | int64 |
| Incidentes_Regras RT | PUNISHMENT_TYPE | Tipo de punição (permanent_block, temporary_block, warning) | object |
| Incidentes_Regras RT | DISCIPLINE_RULE_BUCKET | É uma derivada de um indicador interno da Rappi | object |
| Incidentes_Regras RT | CATEGORY_RULE | Gênero da regra aplicável | object |
| Incidentes_Regras RT | ORDER_ID | ID do pedido | float64 |
| infos gerais | ID | ID do entregador | int64 |
| infos gerais | NOME | Nome | object |
| infos gerais | SOBRENOME | Sobrenome | object |
| infos gerais | GENERO | Gênero | object |
| infos gerais | DATA_NASCIMENTO | Data de Nascimento | object |
| infos gerais | CIDADE | Cidade de atuação | object |
| infos gerais | IS_ACTIVE | Se o entregador está ativo ou não | bool |
| infos gerais | TRANSPORTE | Modal de transporte | object |
| infos gerais | AUTO_ACEITE | Se o entregador aceita pedidos automaticamente | bool |

| | | | |
|----------------------|--------------------------------|---|---------|
| | | | |
| infos gerais | COUNT_ORDERS_LAST_7D | Ordens realizadas nos últimos 7 dias | int64 |
| infos gerais | COUNT_ORDERS_LAST_30D | Ordens realizadas nos últimos 30 dias | int64 |
| infos gerais | COUNT_ORDERS_CANCELED_LAST_7D | Ordens canceladas nos últimos 7 dias | int64 |
| infos gerais | COUNT_ORDERS_CANCELED_LAST_30D | Ordens canceladas nos últimos 30 dias | int64 |
| infos gerais | GORJETA | Gorjetas recebidas pelo entregador | float64 |
| infos gerais | PRIMEIRO_PEDIDO | Data do primeiro pedido entregue | object |
| infos gerais | ULTIMO_PEDIDO | Data do último pedido entregue | object |
| infos gerais | COUNT_ORDERS_RESTAURANTES | Ordens de restaurantes | int64 |
| infos gerais | COUNT_ORDERS_MERCADO | Ordens de mercado | int64 |
| infos gerais | COUNT_ORDERS_FARMACIA | Ordens de farmácia | int64 |
| infos gerais | COUNT_ORDERS_EXPRESS | Ordens turbo (10min) | int64 |
| infos gerais | COUNT_ORDERS_ECOMMERCE | Ordens e-commerce | int64 |
| infos gerais | COUNT_ORDERS_OUTSIDE | Compras em uma loja que não está no aplicativo | int64 |
| infos gerais | FRETE_MEDIO | Média do tempo gasto na entrega dos pedidos | float64 |
| infos gerais | COOKING_TIME_MEDIO | Média de tempo esperando o pedido ficar pronto nos restaurantes | float64 |
| infos gerais | ITENS_MEDIO | Média de itens nas entregas | float64 |
| Ordens Done e Cancel | STOREKEEPER_ID | ID do comerciante | int64 |
| Ordens Done e Cancel | ORDERS_DONE | Número de pedidos realizados | int64 |
| Ordens Done e Cancel | ORDERS_CANCEL | Ordens totais canceladas pelo entregador | int64 |
| Ordens Done e Cancel | CANCEL_OPS_RT | Ordens canceladas manualmente pelo time de operação | int64 |

| | | ID do | |
|---------------------------|----------------------|---|---------|
| Product return | ID_ENTREGADOR | entregador | int64 |
| Product return | LEVEL_NAME | Nível do entregador | object |
| Product return | MODAL | Meio de locomoção | object |
| Product return | CITY | Cidade de atuação | object |
| Product return | CREATED_AT | Data de cadastro | object |
| Product return | ORDER_ID | ID do pedido | int64 |
| Product return | PRODUCT_RETURNS | Valor da devolução (negativo) | float64 |
| Product return | VERTICAL_SUB_GROUP | Categoria do pedido (farmácia, mercado) | object |
| Product return | COUNT_TO_GMV | Abater do GMV | bool |
| Product return | GMV | Gross Merchandise Value | float64 |
| Product return | STORE_ID | ID da loja | int64 |
| supply | SUPPLY_HOURS | ID do entregador | int64 |
| supply | CITY | Cidade de atuação | int64 |
| supply | DATE | Dia da semana (YYYY-MM-DD) | object |
| supply | WEEK | Segunda-feira da semana em referência (YYYY-MM-DD) | object |
| supply | CREATED_CARD | Data de Criação do Cartão | object |
| supply | LEVEL_NAME_2 | Nível do entregador | object |
| supply | HAVE_CARD | Tem um cartão? | bool |
| supply | TRANSPORT_MEDIA_TYPE | Modal de transporte | object |
| supply | NUM_ORDERS | Número de pedidos atendidos pelo entregador naquele dia | int64 |
| supply | SUPPLY_HOURS | Horas em que o entregador ficou conectado no período | float64 |
| tempo resolucao e modal 2 | TICKET_ID | ID do ticket de suporte | object |
| tempo resolucao e modal 2 | STOREKEEPER_ID | ID do entregador que abriu o ticket | int64 |
| tempo resolucao e modal 2 | LEVEL_NAME | Nível do entregador | object |

| | | | |
|---------------------------|--------------------------|-----------------------------------|---------|
| | | | |
| tempo resolucao e modal 2 | TRANSPORT_MEDIA_TY PE | Model de transporte do entregador | object |
| tempo resolucao e modal 2 | SENT_DATA | Data de envio do ticket | object |
| tempo resolucao e modal 2 | SENT_HOUR | Hora de envio do ticket | object |
| tempo resolucao e modal 2 | RESPONSE_AT | Data e Hora de Resposta do ticket | object |
| tempo resolucao e modal 2 | RESPONSE_TIME | Tempo de resposta (em minutos) | int64 |
| tempo resolucao e modal 2 | RESOLUTION_TIME | Tempo de resolução (em minutos) | float64 |
| tempo resolucao e modal 2 | RESOLUTION_TIME_BUCKET | Categoria de tempo de resposta | object |
| tempo resolucao e modal 2 | CITY | Cidade do entregador | object |

Forma de agregação e mesclagem de dados:

Todos os arquivos CSV tem uma coluna com um ID único do entregador (geralmente a coluna tem o nome "STOREKEEPER_ID", mas em alguns CSVs pode ter um nome diferente como "ID_ENTREGADOR"). Desse modo, podemos usar esse identificador único para juntar as diferentes tabelas (fazer um "join") e assim cruzar as informações de um CSV com o outro.

Riscos e contingências relacionados aos dados (qualidade, cobertura/diversidade e acesso):

Em relação a qualidade, temos o risco que os arquivos repassados possam ter dados que não são verossímeis por uma série de motivos (inserções errôneas, importações indevidas, "bugs" no aplicativo ou base de dados). Adicionalmente, podemos verificar um nível elevado de ruído em alguns arquivos CSV, como informações ausentes (campos nulos), muitos outliers inconsistentes (por exemplo, muitos entregadores com receitas além do esperado no arquivo "earnings") e valores repetidos (especialmente no arquivo "criacao contas churn").

Sobre cobertura e diversidade, temos alguns arquivos com muitas informações, como o "criacao contas churn" com mais de 32 milhões de registros. Isso pode levar a um excesso de informações para tratamento e análise, caso não seja feito um cruzamento e filtragem prévios.

Finalmente, no que concerne ao acesso, a Rappi não pode liberar alguns dados, como as regras automáticas usadas para aplicar suspensões e avisos na plataforma (por exemplo, quanto

tempo um entregador pode ficar parado até receber um aviso). Assim, isso pode dificultar algumas análises.

Seleção do subconjunto para análises iniciais:

Temos algumas hipóteses iniciais sobre as causas do "churn", que irão embasar nossas análises iniciais. Identificamos essas hipóteses a seguir, junto com o conjunto de dados relevante:

- *Hipótese 1: Os entregadores ("RT"s) deixam a plataforma por muitas suspensões recorrentes.*
 - Subconjunto relevante: arquivo "Incidentes_Regras RT", com as suspensões. Verificar se existe uma relação entre número e duração de suspensões e "churn".
- *Hipótese 2: Os entregadores deixam a plataforma por devoluções excessivas de pedidos, que levam a dívidas no valor dos produtos e custos com devoluções (gasolina, tempo de deslocamento).*
 - Subconjunto relevante: arquivo "comp defects", com as devoluções pendentes. Verificar se existe uma relação entre devoluções e "churn".
- *Hipótese 3: Os entregadores deixam a plataforma por baixa remuneração ou poucos pedidos.*
 - Subconjunto relevante: arquivos "earnings", "Ordens done e cancel" e, especialmente, "Infos gerais". Verificar se existe uma relação entre rendimentos, número de ordens realizadas e "churn".

Restrições de segurança: foi mencionado que são dados sensíveis e que devem ser geridos com cuidado. Portanto, não vamos publicar essa base de dados, nem repassar essas informações para terceiros. O uso dos dados será exclusivamente para desenvolver um modelo preditivo para a possibilidade de "churn" de entregadores na plataforma Rappi.

4.2.2. Estatística Descritiva dos Dados

Link para o Google Colab com o código das análises:

<https://colab.research.google.com/drive/1s7qv2MbyDceq2NBiZt31mBhXniZ3x8j5?usp=sharing>

A partir dos dados fornecidos, podemos estabelecer algumas análises pautadas em estatística descritiva e elaborar gráficos para testar as hipóteses suscitadas inicialmente.

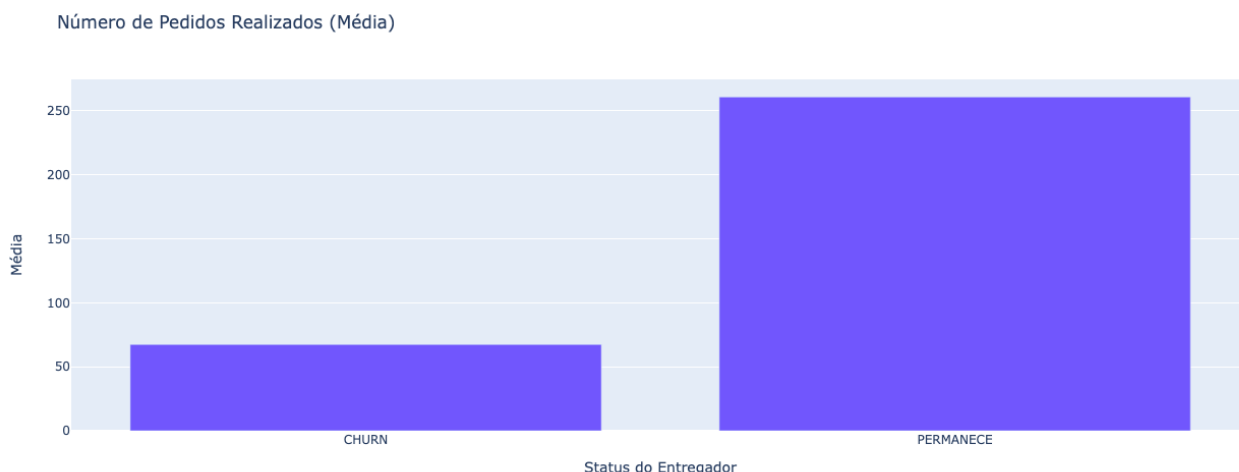
- *Hipótese A: Os entregadores deixam a plataforma por baixa remuneração ou poucos pedidos*

Para fins de comparação para essa hipótese, foi necessário trabalhar com dois grupos – os entregadores que saíram da plataforma e aqueles que permanecem na mesma. Para tanto, dividimos o arquivo "Infos Gerais" em dois grupos, os que entregaram o último pedido há mais de 21 dias ("CHURN")- critério informado em entrevista com a Rappi para classificação de Churn - e os que entregaram pedidos após essa data ("PERMANECE"):

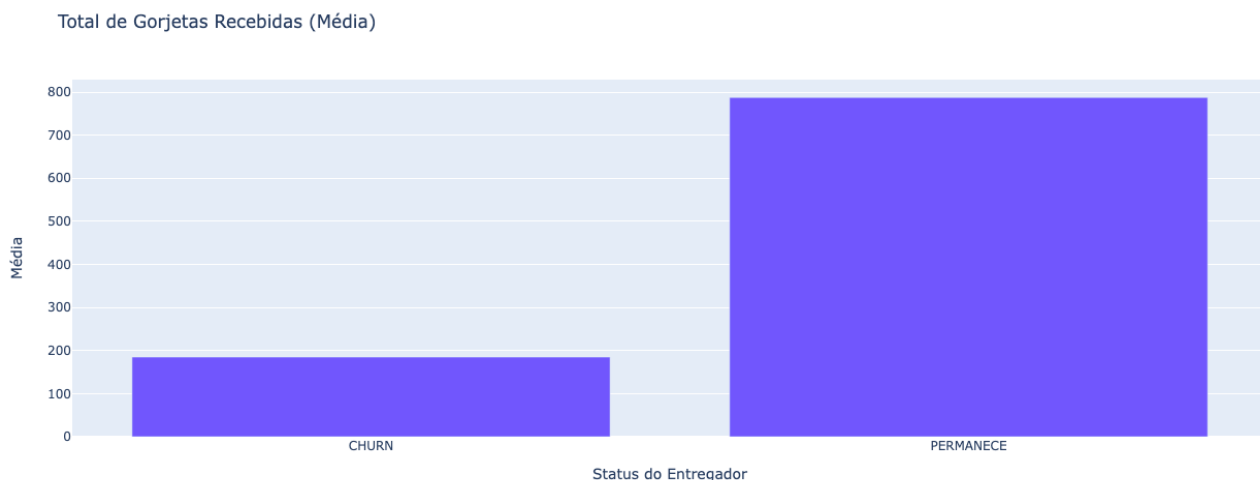
```
df_infos = pd.read_csv('/content/drive/MyDrive/data/infos gerais.csv')
df_infos.sort_values('ULTIMO_PEDIDO', ascending=False)
# Churns são entregadores com o último pedido há mais de 21 dias
# Os dados vão até 01/08/2022, logo a data de corte é 11/07/2022
df_not_churned = df_infos[(df_infos['ULTIMO_PEDIDO'] > '2022-07-11')]
df_churned = df_infos[(df_infos['ULTIMO_PEDIDO'] < '2022-07-11')]
```

Realizando uma comparação entre os dois grupos, obtivemos resultados interessantes.

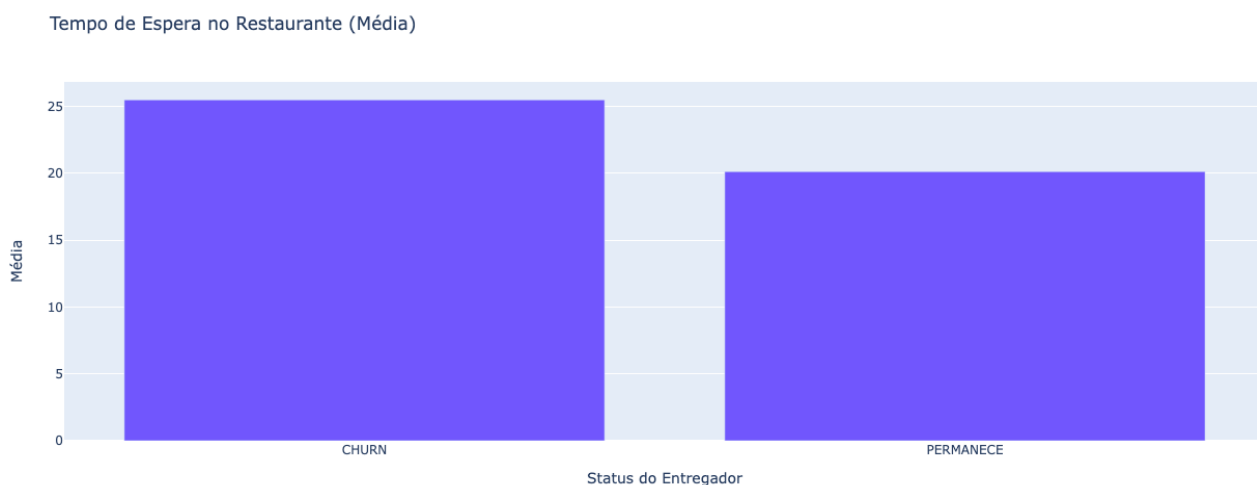
- Os entregadores que sofreram "churn" realizaram menos entregas do que aqueles que permaneceram na plataforma, sendo o número de pedidos cerca de 4 vezes maior. Isso pode ser um indicador de que quanto mais corridas o entregador recebe, menor sua propensão a sair da plataforma:



- Os entregadores que permaneceram na plataforma ganharam quase o quádruplo de gorjetas daqueles que deram "churn". Isso pode indicar que as gorjetas (ou ausência delas) são um fator de permanência na plataforma:

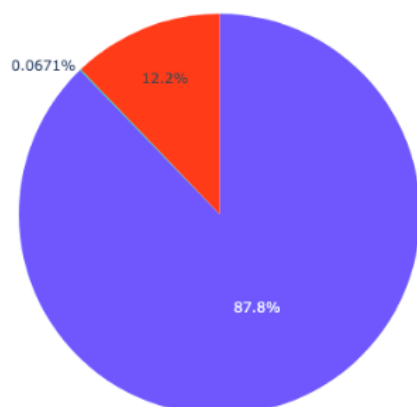


- Entregadores que permaneceram na plataforma em geral esperam menos tempo para o pedido ficar pronto no restaurante. Não conseguimos entender qual a causa desse fator:

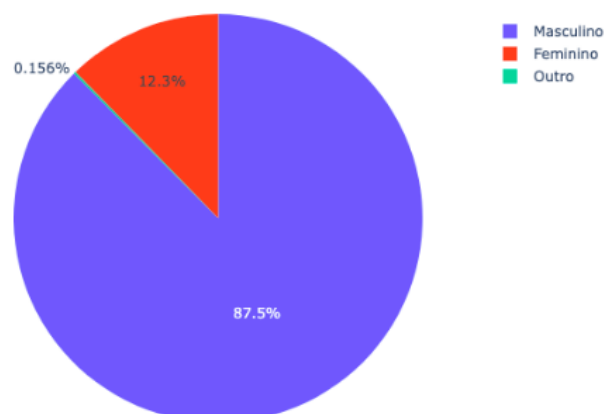


- Aspectos demográficos, como gênero do entregador, aparentam não impactar de forma expressiva o "churn". A composição dos dois grupos é praticamente idêntica:

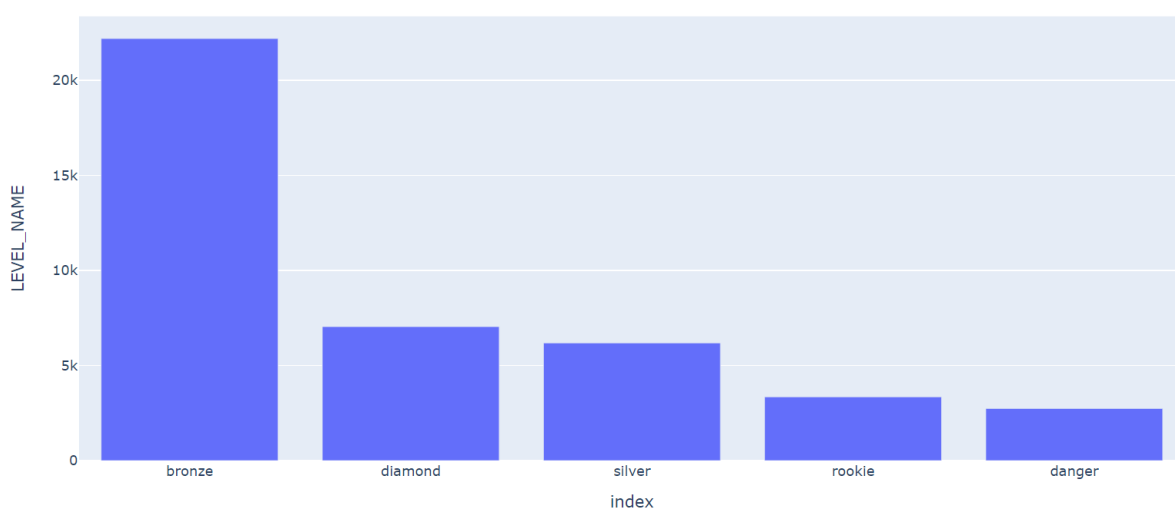
Gênero dos entregadores que deram 'Churn'

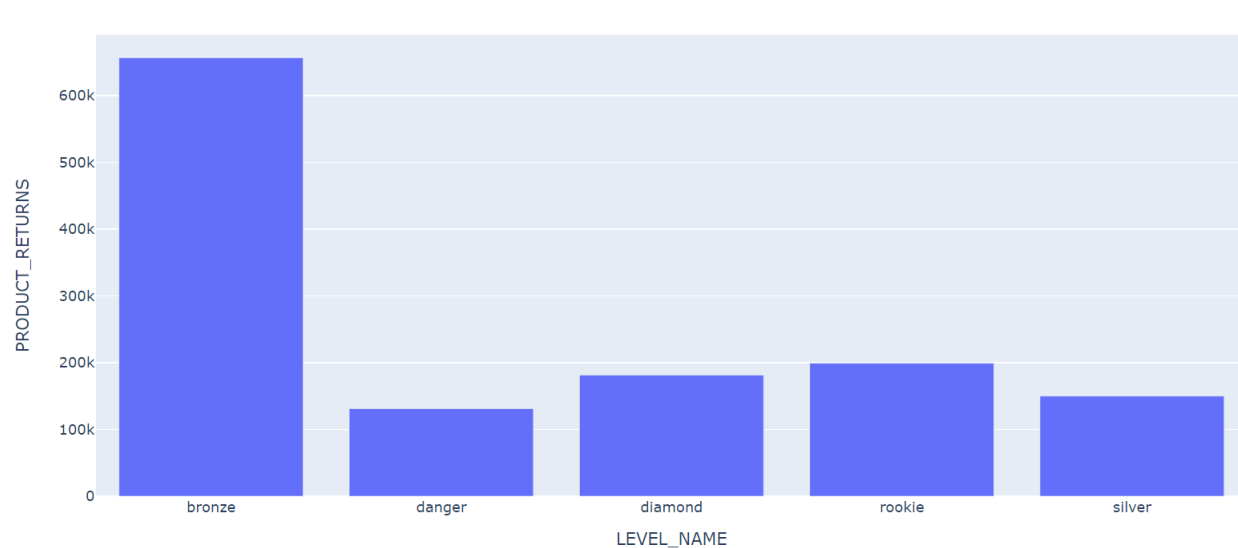


Gênero dos entregadores que permaneceram na plataforma



- *Hipótese B: Os entregadores deixam a plataforma por devoluções excessivas de pedidos*
- Para trabalhar com essa hipótese, verificamos se algum grupo está representado de forma desproporcional no valor de pedidos a devolver pendentes. Assim, comparamos a quantidade de entregadores em cada nível e a quantidade de devoluções pendentes:





Nesse contexto, verificamos que há uma quantidade expressiva de devolução no nível bronze (porém proporcional), e uma quantidade muito acima do esperado no nível "rookie" e "danger". O que mais se destaca nessa análise é que o nível "bronze" pode representar um grupo descontente com a plataforma, dado o alto volume de devoluções pendentes.

4.2.3. Descrição da Predição Desejada ("Target")

A predição desejada ("target") consiste em classificar se um entregador ("RT") irá sair da plataforma Rappi (evento de "churn") ou se irá permanecer na plataforma.

Diferentemente de um modelo de regressão contínua, em que se deseja encontrar um valor numérico específico por interpolação (por exemplo, o valor de uma casa em determinado mercado imobiliário), no modelo de classificação o mais importante é prever em qual categoria se deve encaixar uma situação (por exemplo, um paciente tem uma patologia ou não tem).

Nosso caso, portanto, consiste em um modelo de **classificação binária**, em que o "target" pode ser encaixado em uma de duas categorias: a) o entregador saiu; ou b) permaneceu na plataforma durante determinado período.

4.3. Preparação dos Dados

- Link para o notebook de "*feature engineering*":

https://colab.research.google.com/drive/1gcgucA_kH5xj9dncwh1sUfaLRR0k3G?usp=sharing

Após cuidadosa análise dos dados disponibilizados pelo cliente, iniciamos a preparação e também a seleção das características, propriedades e atributos ("features") que serão utilizadas em nosso modelo preditivo. Nesta fase não nos limitamos em apenas buscar features prontas, mas também criamos novas features com base nos dados fornecidos para que pudéssemos explicitar informações valiosas, mas que não estavam visíveis nos dados.

4.3.1. Manipulação

Importamos as bases de dados em formato ".CSV" para nosso ambiente de desenvolvimento ("Google Colaboratory") e realizamos algumas otimizações para conseguir lidar com problemas gerados pelo tamanho das bases. As principais mudanças que realizamos foram:

- mudança de tipo de algumas variáveis, como por exemplo de int64 para int16 em algumas variáveis que não armazenam grandes valores que exigissem, portanto, grande quantidade de memória volátil ("RAM");
- mudança das variáveis categóricas de 'object' para 'category', também para economizar memória RAM;
- encoding das variáveis categóricas usando Label-Encoding (se ordinais) e One-Hot Encoding (se nominais);
- mudança de 'object' para 'datetime' nas colunas que representam datas;
- transformação dos 'datetimes' em 'ordinais', para introdução no modelo;
- renomeação de algumas colunas para nomes mais intuitivos.

4.3.2. Agregação

Para juntar as bases de dados realizamos algumas operações de merge através da biblioteca "pandas" da linguagem Python. Utilizamos a operação de Inner Join - que retorna um dataframe com apenas as linhas que estão presente em ambos os dataframes, concatenados através de um valor de referência (no caso, o ID dos entregadores). Utilizamos também a operação de Left Join - que retorna um dataframe com todas as linhas dos primeiro dataframe que está sendo concatenado.

Além disso, aplicamos a função "*groupby*" para conseguirmos:

- agrupar os ganhos e as gorjetas de cada entregador;

- somar os minutos de punição de cada entregador, de forma que conseguíssemos visualizar o tempo total de punição de cada ID (entregador) da base;
- realizar uma contagem, por entregador, de pedidos em aberto para o suporte, a soma de devoluções que constam como pendentes na base e, por fim, uma contagem de pedidos com problema por entregador.

4.3.3. Remoção e Substituição

Retirar entregadores sem earnings, Inner Join em 'earnings', 'ordens realizadas e canceladas', 'média de km rodados', 'taxa de aceitação dos pedidos'.

Fizemos também algumas operações de remoção e substituição nas bases. Através de operações de Inner join - comentadas previamente no tópico 4.3.2 - juntamos os dataframes que continham as informações de earnings, ordens realizadas e canceladas, média de km rodados e taxa de aceitação dos pedidos – removendo, assim, qualquer entregador que não possuísse qualquer um desses atributos.

Na tabela de 'taxa de aceitação dos pedidos', alguns entregadores tinham valores em branco como porcentagem de aceitação. Realizamos uma operação 'dropna' para eliminar esses entregadores.

Em outras tabelas, realizamos operações de Left Join, realizando as substituições adequadas para os valores ausentes na tabela à direita na operação de merge com o método "*fillna*" do Pandas:

- "Incidentes Regras RT" (Medidas Disciplinares contra Entregadores) – se o entregador não tinha qualquer registro na tabela de medidas disciplinares, substituímos o valor vazio de "PUNISHMENT_MINUTES" por "0";
- "Quantidade de pedidos ao suporte da Rappi"- se o entregador não tinha qualquer registro na tabela de pedidos de suporte, substituímos o valor vazio de "SUPPORT_TICKET_COUNT" contagem por "0";
- "Devoluções Pendentes" – se o entregador não tinha qualquer devolução pendente na tabela, substituímos o valor "PRODUCT_RETURNS" por "0";
- "Pedidos com Problema" – se o entregador não tinha qualquer pedido com problema em seu ID, substituímos o valor "DEFECTS_COUNT" por "0".

4.3.4. Seleção das features

| Nome da feature escolhida | Motivo |
|---------------------------|---|
| LEVEL_NAME | É relevante, pois após o merge com outras tabelas, nos permite verificar se contas com níveis mais baixos dão mais churn. |

| | | |
|--------------------------------|-----------|--|
| | TRANSPORT | É relevante, pois conseguimos verificar se há algum modal em que os entregadores dão mais churn |
| AUTO_ACCEPT | | É relevante, pois conseguimos verificar se os entregadores sem "auto aceite" dão mais churn. |
| ACCEPTANCE_RATE | | É relevante, pois conseguimos realizar um agrupamento por STOREKEEPER_ID, nos permitindo verificar se entregadores que rejeitam mais pedidos dão mais churn. |
| ORDERS_DONE | | É útil pois, ao contabilizar quantas entregas foram feitas por um RT, podemos observar se o volume de entregas tem relação com o churn. |
| ORDERS_LAST_30D_COUNT | | É relevante pois podemos utilizar para contabilizar as entregas nos últimos dias, comparar com a taxa de churn. |
| ORDERS_CANCEL | | É útil pois, ao contabilizar quantas entregas de um RT foram canceladas, podemos observar se o volume de cancelamentos tem relação com o churn. |
| ORDERS_CANCELED_LAST_30D_COUNT | | É relevante pois podemos contabilizar a quantidade de cancelamentos nos últimos 30 dias, e observar se os entregadores com mais cancelamentos tendem a dar churn. |
| FIRST_ORDER_DATE | | É relevante pois podemos usar a data do primeiro pedido e de último para calcular o tempo de atividade do RT, e relacionar com o churn. |
| LAST_ORDER_DATE | | Bem como o último pedido, é relevante pois podemos usar a data do primeiro pedido e de último pedido para calcular o tempo de atividade do RT, e relacionar com o churn. |
| EARNINGS | | É relevante, pois conseguimos realizar um agrupamento por STOREKEEPER_ID, nos permitindo verificar se entregadores com menos receita dão mais churn. |
| TIPS | | É relevante, pois conseguimos realizar um agrupamento por STOREKEEPER_ID, nos permitindo verificar se entregadores com menos gorjetas dão mais churn. |
| PUNISHMENT_MINUTES | | É relevante, pois conseguimos realizar um agrupamento por STOREKEEPER_ID, nos permitindo verificar se entregadores com mais minutos de punição dão mais churn. |
| PRODUCT_RETURNS | | É o valor da devolução em um pedido cancelado, pode ser útil para visualizarmos se tem alguma relação na devolução de compras de maior volume, com o churn. |

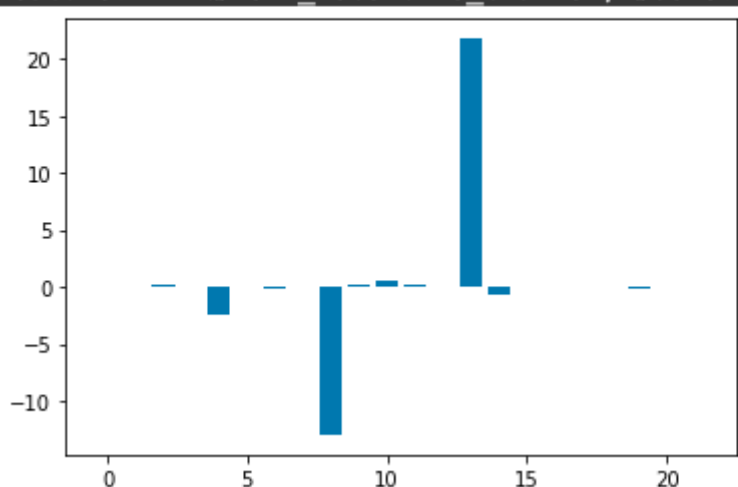
| | |
|----------------------|---|
| SUPPORT_TICKET_COUNT | É relevante, pois conseguimos realizar um agrupamento por STOREKEEPER_ID, nos permitindo verificar se os entregadores que abrem mais "chamados" para o suporte da Rappi dão mais churn. |
| DEFECTS_COUNT | É relevante, pois conseguimos realizar um agrupamento por STOREKEEPER_ID, nos permitindo verificar se os entregadores com mais "ordens problemáticas" dão mais churn. |
| AVG_DISTANCE_TO_USER | É relevante, pois conseguimos realizar um agrupamento por STOREKEEPER_ID, nos permitindo entender se entregadores com pedidos mais longos dão mais churn. |

Ao invés de realizarmos uma análise gráfica exploratória para cada uma das features acima, optamos por construir um modelo preliminar usando regressão logística (disponível em <https://colab.research.google.com/drive/1kowwKCfPTIs6mnsk5RbLtMWLaypdAG3D#scrollTo=OJyQcGeaRkWg>) para demonstrar como podemos apresentar as hipóteses mais pertinentes através de um modelo. Os resultados obtidos foram os seguintes:

```

Feature: LEVEL_NAME, Score: 0.05326
Feature: AUTO_ACCEPT, Score: 0.06191
Feature: ACCEPTANCE_RATE, Score: 0.12545
Feature: ORDERS_DONE, Score: -0.06278
Feature: ORDERS_LAST_30D_COUNT, Score: -2.50861
Feature: ORDERS_CANCEL, Score: 0.07066
Feature: ORDERS_CANCELED_LAST_30D_COUNT, Score: -0.13663
Feature: FIRST_ORDER_DATE, Score: -0.02853
Feature: LAST_ORDER_DATE, Score: -12.97138
Feature: EARNINGS, Score: 0.22665
Feature: TIPS, Score: 0.48481
Feature: PUNISHMENT_MINUTES, Score: 0.25026
Feature: PRODUCT_RETURNS, Score: 0.04352
Feature: SUPPORT_TICKET_COUNT, Score: 21.75265
Feature: DEFECTS_COUNT, Score: -0.72360
Feature: AVG_DISTANCE_TO_USER, Score: 0.02992
Feature: TRANSPORT_bicycle, Score: 0.01427
Feature: TRANSPORT_car, Score: -0.02671
Feature: TRANSPORT_motorbike, Score: 0.00896
Feature: TRANSPORT_neither, Score: -0.11032
Feature: TRANSPORT_cargo_van, Score: 0.00927
Feature: TRANSPORT_motorbike_trailer, Score: 0.00000

```



No gráfico acima, quanto *mais negativo* o valor da feature em questão, *menos* propenso um entregador é de dar "churn" se tiver um valor alto naquela feature. Inversamente, quanto *mais positivo* o valor da feature no gráfico acima, *mais propenso* um entregador é de dar "churn" se tiver um valor alto naquela feature.

No caso, se observa um valor muito negativo para "LAST_ORDER_DATE", ou seja, quanto mais tempo um entregador fica sem entregar um pedido, mais cresce sua chance de dar "churn". Isso pode representar um viés, porém, pois uma das definições de "churn" que aplicamos é a de um entregador que está há mais de 21 dias sem realizar entregas.

Outro ponto de destaque é o valor muito positivo para "SUPPORT_TICKET_COUNT". Isso representa o número absoluto de chamados abertos por um entregador para o suporte da Rappi. Todavia, entendemos que esse dado pode representar um ruído, pois em uma análise manual notamos que os entregadores que não deram "churn" estão sem *nenhum* ticket aberto para o suporte, o que é um forte indício de que se trata de um dado enviesado:

```
df_infos.groupby('IS_CHURN')['SUPPORT_TICKET_COUNT'].describe()
```

| | count | mean | std | min | 25% | 50% | 75% | max |
|----------|----------|-----------|-----------|-----|-----|-----|------|--------|
| IS_CHURN | | | | | | | | |
| False | 31133.0 | 0.000000 | 0.000000 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| True | 132160.0 | 12.544597 | 31.301608 | 0.0 | 0.0 | 2.0 | 12.0 | 1184.0 |

4.4. Modelagem

Para o teste de todo modelo preditivo, os dados foram divididos em duas parcelas: dados de treino, e dados de teste, em uma proporção de 70% e 30%, respectivamente.

Realizamos um “undersampling” durante o processo de modelagem com a biblioteca “imbalanced-learn”, pois o conjunto de dados de entregadores que deram “churn” era consideravelmente superior ao dos que permaneceram na plataforma. Esse tratamento essencialmente consiste em eliminar parte do conjunto com dados em excesso para equilibrá-lo com o de dados sub representados. Caso esse tratamento do desbalanceamento identificado não fosse realizado, isso poderia enviesar as métricas apresentadas a seguir.

Excluimos as features 'ORDERS_CANCELED_LAST_30D_COUNT', 'ORDERS_LAST_30D_COUNT' e 'LAST_ORDER_DATE', pois apresentam alta correlação com o target e poderiam enviesar o modelo ao fornecer dicas muito fortes de um entregador prestes a dar churn. Um entregador que realizou poucas ordens nos últimos 30 dias ou ficou muitos dias sem realizar uma ordem seria classificado como churn extremamente provável e traria um viés excessivo para o modelo. Outra feature removida foi 'SUPPORT_TICKET_COUNT', pois por algum motivo os entregadores que não deram churn estavam com essa métrica zerada, o que também traz um viés excessivo para o modelo.

Passada essa análise preliminar, utilizamos, majoritariamente, quatro métricas de avaliação, cada uma tendo sua peculiaridade. Vale ressaltar que esses indicadores se baseiam na “matriz de confusão”, uma matriz que compara as predições falsas e positivas, com o dado real do target.

| | | Detectada | |
|------|-----|-----------------------------|-----------------------------|
| | | Sim | Não |
| Real | Sim | Verdadeiro Positivo (VP) | Falso Negativo (FN) |
| | Não | Falso Positivo (FP) | Verdadeiro Negativo (VN) |

Matriz de confusão

Acurácia: Indica a performance geral do modelo, apresentando uma porcentagem de quantas classificações o modelo classificou corretamente.

Precisão: Indica quantos dos positivos feitos pelo modelo foram realmente assertivos.

Revocação/Recall: Indica quantos dos positivos dados pelo modelo foram verdadeiros ou falsos.

F1 Score: É a média harmônica entre precisão e recall ($2 * \text{precisão} * \text{recall} / (\text{precisão} + \text{recall})$).

Para fins de aprofundamento, as fórmulas para cálculo de cada uma das métricas já foram mencionadas na seção 4.1.3.

4.4.1. Avaliação e Seleção de Modelos (Algoritmos)

Dentre as quatro principais métricas de avaliação de modelos, demos prioridade a duas, acurácia e recall. Basicamente, buscamos um modelo com uma alta taxa de acertos, por isso priorizamos a acurácia. Também desejamos o menor número possível de falso negativos, por valorizamos a revocação ("recall") sobre a precisão, dado que é mais importante que o modelo identifique todos os entregadores propensos a dar churn ("recall") do que prevenir a identificação incorreta como churn um entregador que na realidade não é churn ("precisão").

Para termos um sentido inicial dos modelos que poderiam apresentar as melhores métricas, usamos a biblioteca Pycaret. Essa ferramenta possibilita rodar uma série de modelos de classificação e observar as métricas de cada um, para que possamos escolher o modelo mais adequado para implementar manualmente.

Atualmente, após o teste com vários algoritmos, elegemos os quatro melhores em termos de acurácia, sendo eles:

1. *Light Gradient Boosting Machine (LightGBM);*
2. *Random Forest Classifier;*
3. *Gradient Boosting Classifier; e*
4. *Ada Boost Classifier.*

No caso específico do modelo Extra Trees Classifier, optamos por excluí-lo pelo fato de ser muito semelhante ao Random Forest. Abaixo estão as métricas de cada modelo, com a Acurácia, Revocação, Precisão e F1 Score.

A tabela abaixo contém os modelos de machine learning aplicados ao nosso "dataset" pela ferramenta Pycaret, em ordem decrescente de acurácia:

| Modelo | Acurácia | Revocação | Precisão | F1 Score |
|--------|----------|-----------|----------|----------|
|--------|----------|-----------|----------|----------|

| Light Gradient Boosting Machine | | | | |
|---------------------------------|--------|--------|--------|--------|
| | 0.9040 | 0.9398 | 0.9434 | 0.9415 |
| Random Forest Classifier | 0.9016 | 0.9371 | 0.9429 | 0.9400 |
| Extra Trees Classifier | 0.9014 | 0.9379 | 0.9420 | 0.9399 |
| Gradient Boosting Classifier | 0.8750 | 0.8905 | 0.9544 | 0.9213 |
| Ada Boost Classifier | 0.8660 | 0.8800 | 0.9534 | 0.9152 |
| Decision Tree Classifier | 0.8589 | 0.8958 | 0.9300 | 0.9126 |
| K Neighbors Classifier | 0.8163 | 0.8261 | 0.9435 | 0.8809 |
| SVM - Linear Kernel | 0.8106 | 0.8154 | 0.9470 | 0.8761 |
| Logistic Regression | 0.8100 | 0.8135 | 0.9481 | 0.8757 |
| Ridge Classifier | 0.8074 | 0.8140 | 0.9441 | 0.8743 |
| Linear Discriminant Analysis | 0.8074 | 0.8140 | 0.9441 | 0.8743 |
| Naive Bayes | 0.2102 | 0.0409 | 0.9706 | 0.0785 |
| Quadratic Discriminant Analysis | 0.1776 | 0.0000 | 0.0000 | 0.0000 |
| Dummy Classifier | 0.1776 | 0.0000 | 0.0000 | 0.0000 |

O notebook que usamos para obter a tabela acima pode ser encontrado no link a seguir:

https://colab.research.google.com/drive/1_l8X6shyH90NFIJso10Mw7Lq_NReoq6m#scrollTo=58ZtXVvXFGas

É importante observar que as métricas apresentadas no teste automatizado são ligeiramente diferentes da seção 4.5 com os modelos feitos manualmente, pois os "datasets" de treino e teste são sorteados aleatoriamente e, portanto, irão divergir. Além disso, a biblioteca Pycaret realiza alguns ajustes que optamos por fazer de forma diferente quando fizemos a construção dos modelos de forma manual.

Apesar disso, essa é uma excelente forma de avaliar quais modelos apresentam os resultados mais promissores e serviu para embasar a seleção dos modelos para avaliação manual.

4.4.2. Configuração de Hiperparâmetros

LightGBM

LightGBM é uma estrutura de aumento de gradiente que usa uma série de árvores de decisão para embasar suas escolhas ("ensemble"). Ele pode ser usado para tarefas de classificação ou outras tarefas de machine learning.

Selecionamos os seguintes hiperparâmetros, baseados na documentação disponível em [Parameters Tuning – LightGBM 3.3.2.99 documentation](#):

```
'learning_rate', 'n_estimators', 'num_leaves', 'max_depth', 'min_child_samples', 'subsample',  
'colsample_bytree', 'reg_alpha', 'reg_lambda', 'min_split_gain', 'min_child_weight',  
'subsample_freq', 'max_bin', 'cat_smooth', 'cat_l2', 'max_cat_to_onehot', 'cat_l2',  
'cat_smooth', 'max_cat_threshold', 'metric', 'n_jobs', 'random_state'.
```

Random Forest Classifier

Random Forest é um algoritmo de aprendizado de máquina "ensemble", que usa o resultado de diversas árvores de decisão para embasar suas decisões.

Selecionamos os seguintes hiperparâmetros, baseados no seguinte post: [Random Forest Hyperparameter Tuning in Python | Machine learning](#)

```
'n_estimators', 'max_features', 'criterion', 'max_depth', 'min_samples_split',  
'min_samples_leaf'
```

Gradient Boosting Classifier

Gradient Boosting, assim como LightGBM, é um algoritmo de aumento de gradiente baseado em árvores de decisão. Contudo, o LightGBM é comparativamente mais eficiente.

Selecionamos os seguintes hiperparâmetros, baseados no seguinte post: [Gradient Boosting | Hyperparameter Tuning Python](#)

```
'loss', 'learning_rate', 'n_estimators', 'subsample', 'min_samples_split', 'min_samples_leaf',  
'min_weight_fraction_leaf', 'max_depth', 'min_impurity_decrease', 'init', 'max_features',  
'random_state'
```

AdaBoost

AdaBoost é uma sigla para "Adaptive Boosting". É um algoritmo de aprendizado de máquina muito usado para tarefas de classificação binária. Ele usa uma série de algoritmos de aprendizado mais "fracos" e consolida seus resultados para obter uma decisão mais robusta.

Selecionamos os seguintes hiperparâmetros, baseados no seguinte post: [Tuning of Adaboost with Computational Complexity I by Srijani Chaudhury I Medium](#)

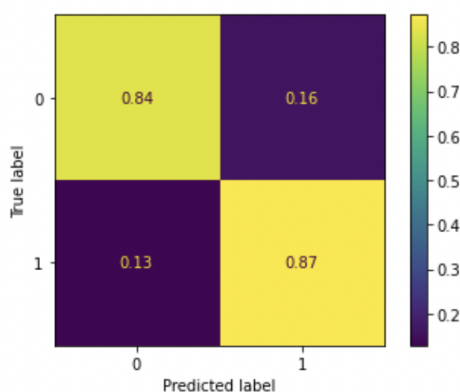
'n_estimators', 'learning_rate', 'algorithm', 'random_state'

4.4.3. Resultados Obtidos

Após selecionarmos os hiperparâmetros para cada modelo como citado na seção anterior, utilizamos o RandomSearch para realizar o tuning dos modelos, achando quais seriam os pontos ótimos de cada hiperparâmetro escolhido. Os resultados obtidos rodando os modelos com as melhores configurações dos hiperparâmetros foram os seguintes:

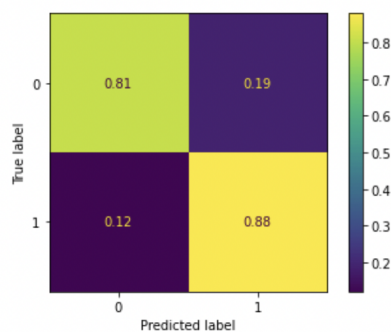
LightGBM

Acc treino: 91,26%
Acc teste: 86,51%
Revocação: 87,12%
Precisão: 96,13%
F1_score: 91,40%



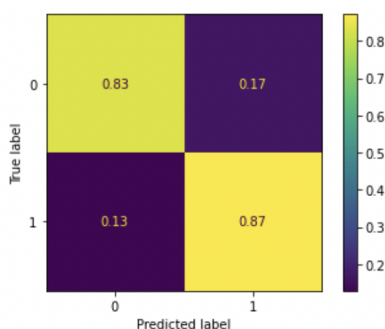
Random Forest Classifier

Acc treino: 85,57%
Acc teste: 86,71%
Revocação: 87,95%
Precisão: 95,56%
F1_score: 91,59%



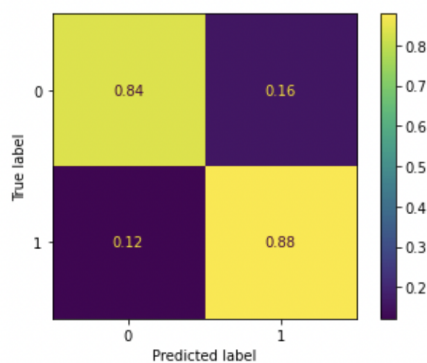
Gradient Boosting Classifier

Acc treino: 87,49%
Acc teste: 86,47%
Revocação: 87,16%
Precisão: 96,04%
F1_score: 91,38%



AdaBoost

Acc treino: 91,26%
Acc teste: 86,51%
Revocação: 87,80%
Precisão: 96,17%
F1_score: 91,80%



4.5. Avaliação

A partir dos resultados obtidos na seção 4.4 com a modelagem e testes preliminares, pudemos selecionar os modelos mais promissores para avaliação e implementação manual, no notebook linkado a seguir:

<https://colab.research.google.com/drive/1kowwKCfPTIs6mnsk5RbLtMWLaypdAG3D#scrollTo=OJyQcGeaRkWg>

Treinamos cada um dos modelos manualmente e verificamos para cada uma:

1. acurácia de treino e de teste;
2. revocação;
3. precisão;
4. F1 score;
5. matriz de confusão; e
6. importância de cada feature para o modelo.

Dessa forma, podemos verificar a qualidade dos resultados obtidos e a taxa de erro, constatando se os resultados preliminares do Pycaret estão de acordo com os modelos do scikit-learn.

Observa-se que os algoritmos escolhidos na seção 4.4. foram adequados pois são todos destinados para tarefas de *classificação* – que é o problema de negócio enfrentado pelo cliente, como apontado na seção 4.1.3. A própria biblioteca que usamos para testes preliminares (Pycaret) apenas usa modelos relacionados com essa tarefa, como pode ser visto na linha de importação do módulo relevante (`from pycaret.classification import *`).

Sendo assim, vamos passar a avaliar os modelos e seus resultados.

4.5.1. Análise dos Resultados

LightGBM

Esse modelo retornou em geral métricas muito boas e deu bastante importância para as features de ordens canceladas e defeitos nas ordens.

```
Acc treino: 0.8752060336646521
Acc teste: 0.8705572619569987
```

```

Revocação: 0.8778389626208541
Precisão: 0.9616111636181687
F1_score: 0.9178174871652007

```

Matriz de confusão:

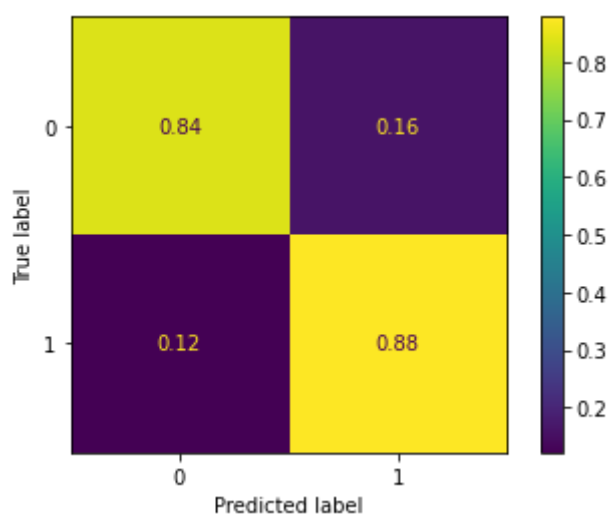
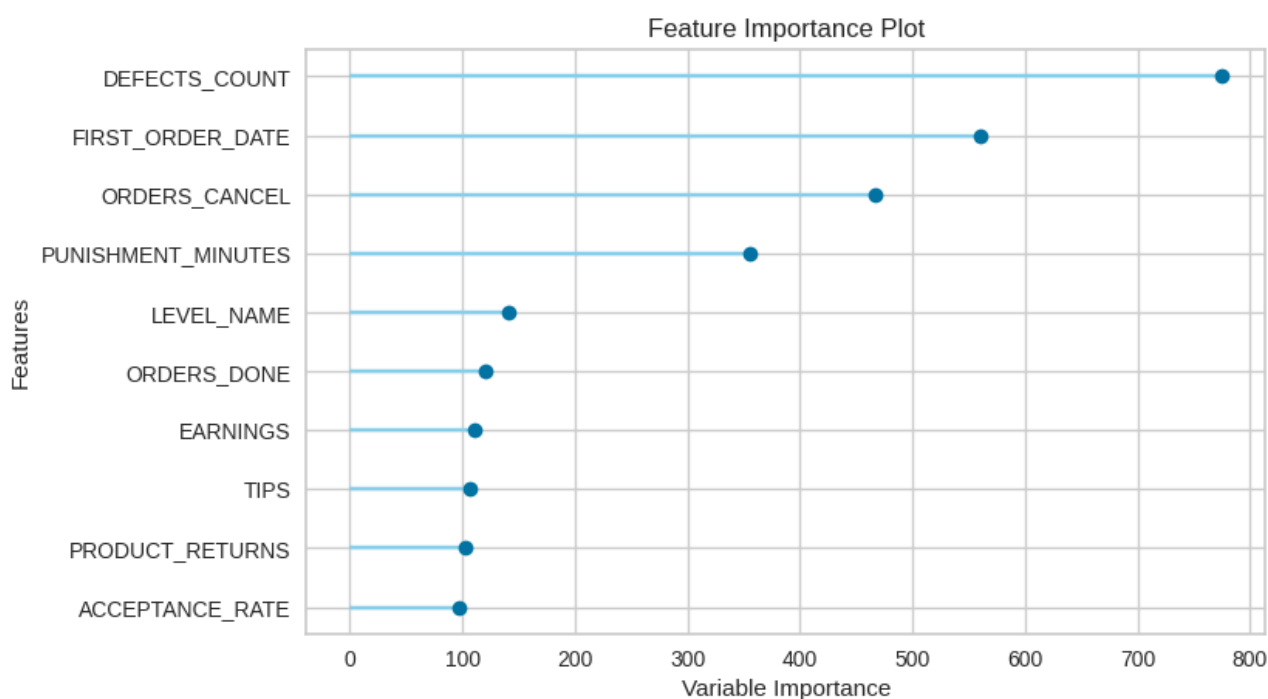
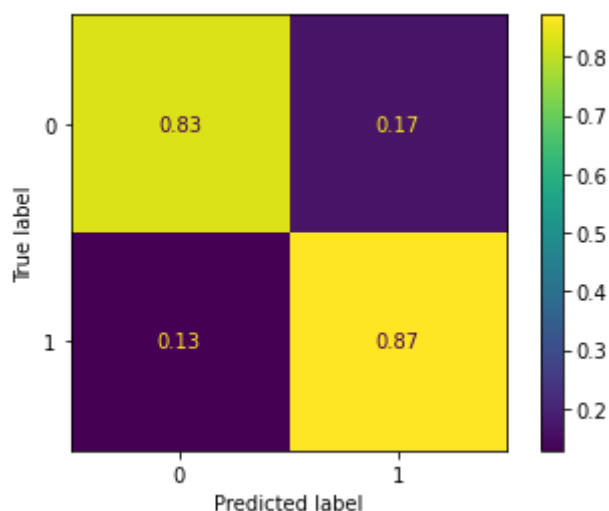


Gráfico de importância das features:



Após o ajuste dos hiperparâmetros, tivemos os seguintes resultados:

```
Acc treino: 0.9126666999650367
Acc teste: 0.8651455316659353
Revocação: 0.8712665262516812
Precisão: 0.9613327733445332
F1_score: 0.914086419424403
```



Random Forest Classifier

O modelo apresentou ótima acurácia, especialmente no dataset de teste. Também apresentou a melhor revocação. Porém, a precisão e acurácia no dataset de treino foram inferiores ao do LightGBM.

```
Acc treino: 1.0
Acc teste: 0.8644769008963831
```

```
Revocação: 0.8715710406780521
Precisão: 0.9601632607419418
F1_score: 0.9137247599031632
```

Matriz de confusão:

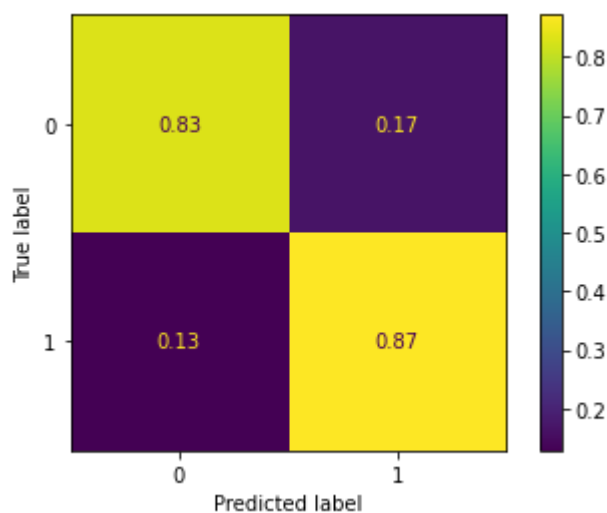
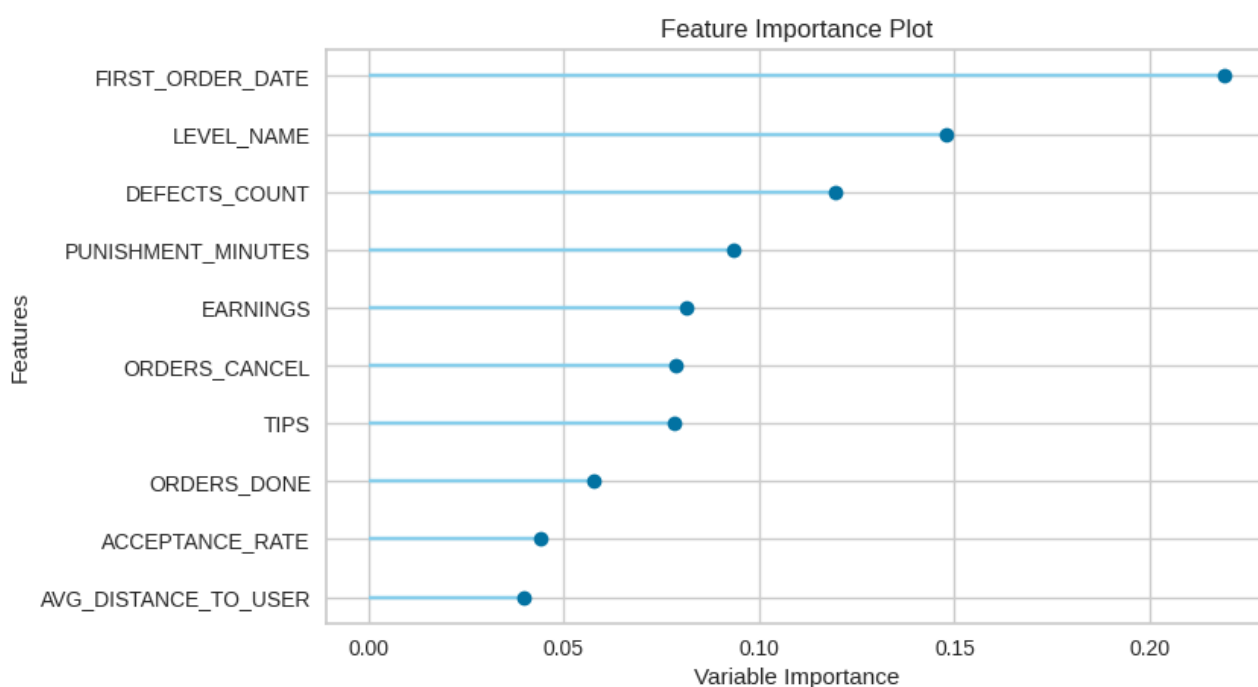
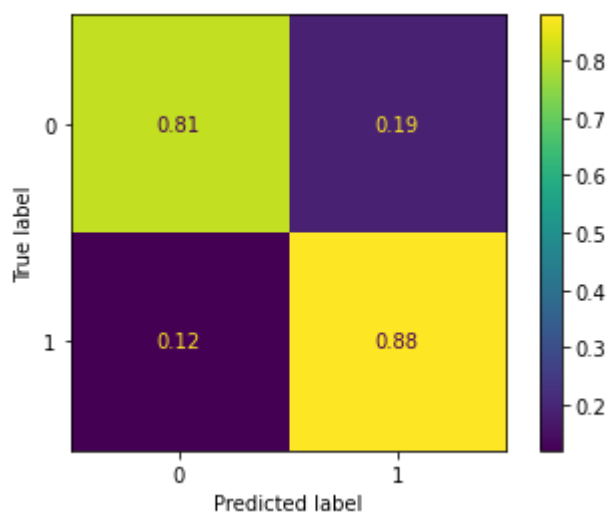


Gráfico de Importância das Features:



Após o ajuste dos hiperparâmetros, tivemos os seguintes resultados:

```
Acc treino: 0.8557514609659858
Acc teste: 0.8671723186861405
Revocação: 0.879539168168092
Precisão: 0.9556106975461814
F1_score: 0.9159982557448102
```



Gradient Boosting Classifier

Os resultados obtidos entre o LightBGM e Gradient Boosting Classifier foram muito semelhantes,, com exceção da acurácia de treino, que foi significativamente superior no LightBGM.

```
Acc treino 0.8538035063183658
Acc teste 0.861133747048622
```

```
Revocação 0.8671048290912782
Precisão 0.9603979875769415
F1 Score 0.9113701224228522
```

Matriz de confusão:

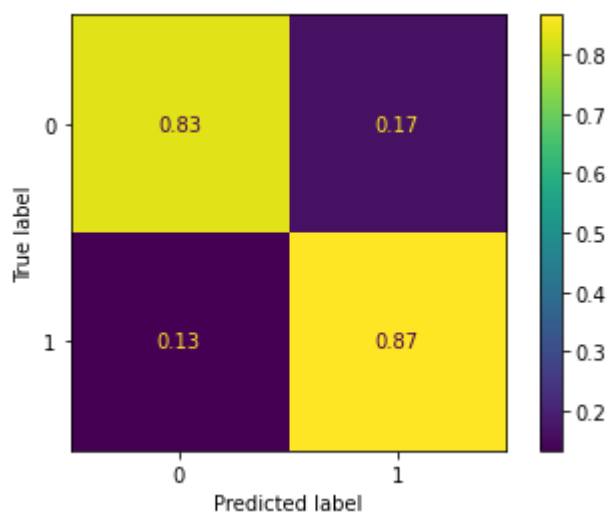
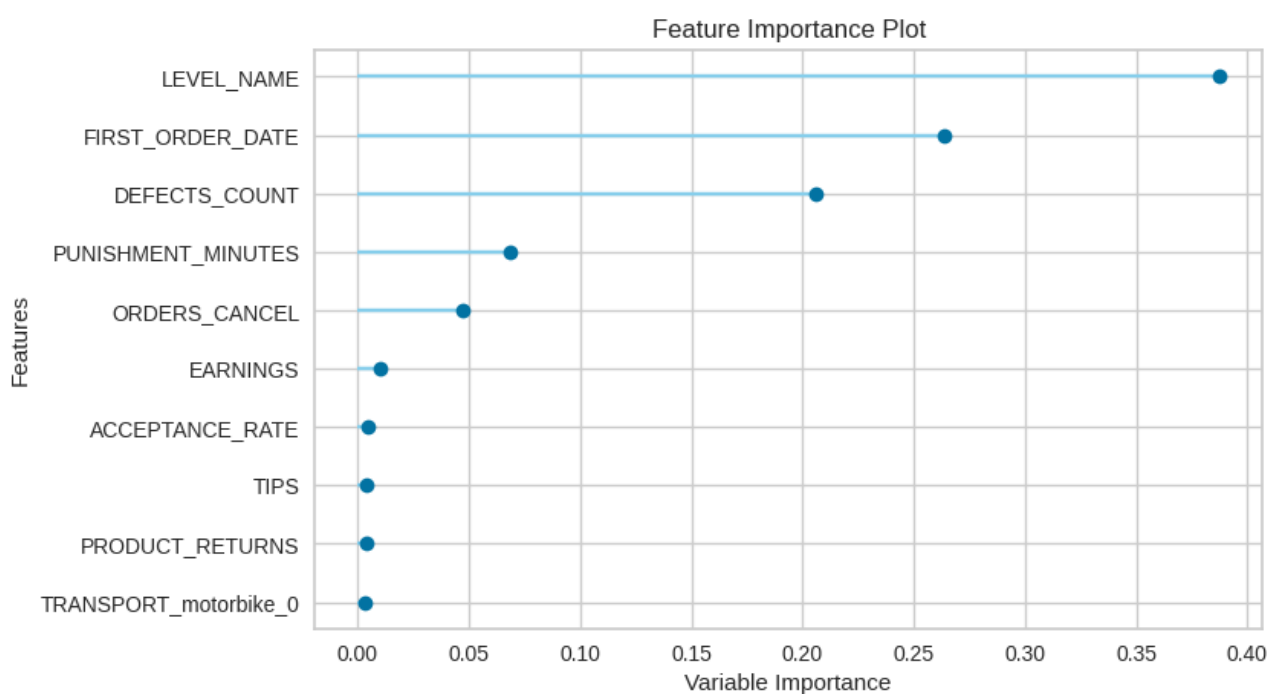


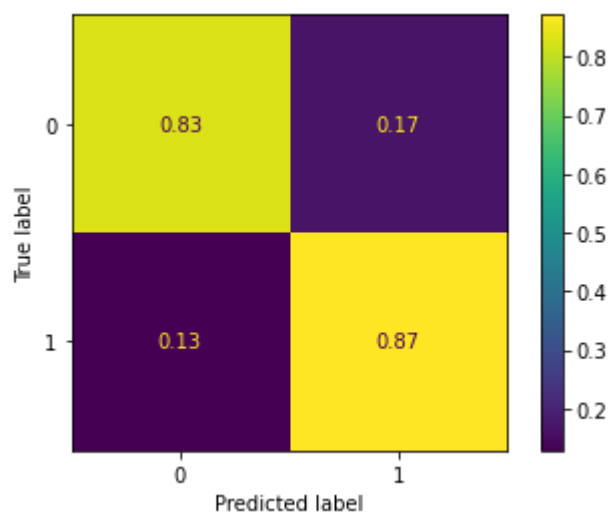
Gráfico de importância das features:



Após o ajuste dos hiperparâmetros, tivemos os seguintes resultados:

```

Acc treino: 0.8749313221117826
Acc teste: 0.8647067427234167
Revocação: 0.8716217930824472
Precisão: 0.9604071132982888
F1_score: 0.9138630588923919
  
```



AdaBoost

Esse algoritmo apresentou boas métricas, porém elas foram inicialmente inferiores às dos demais algoritmos testados. Após o ajuste dos hiperparâmetros, as métricas melhoraram consideravelmente.

```
Acc treino: 0.8449378152939414
Acc teste: 0.8522952840636036
```

```
Revocação: 0.8563453193595046
Precisão: 0.9599476588723901
F1_score: 0.905191722214026
```

Matriz de confusão:

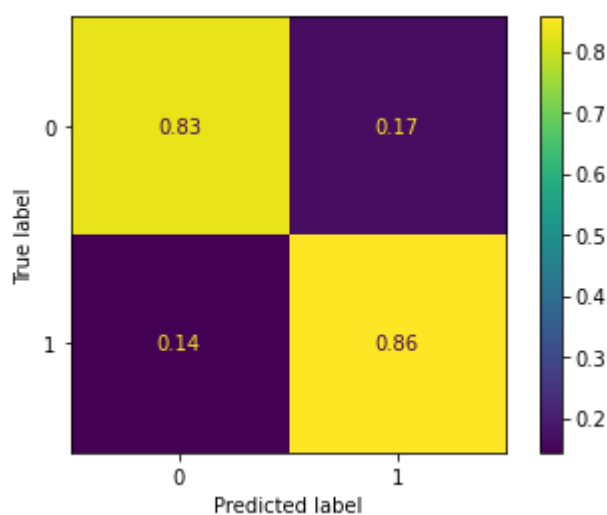
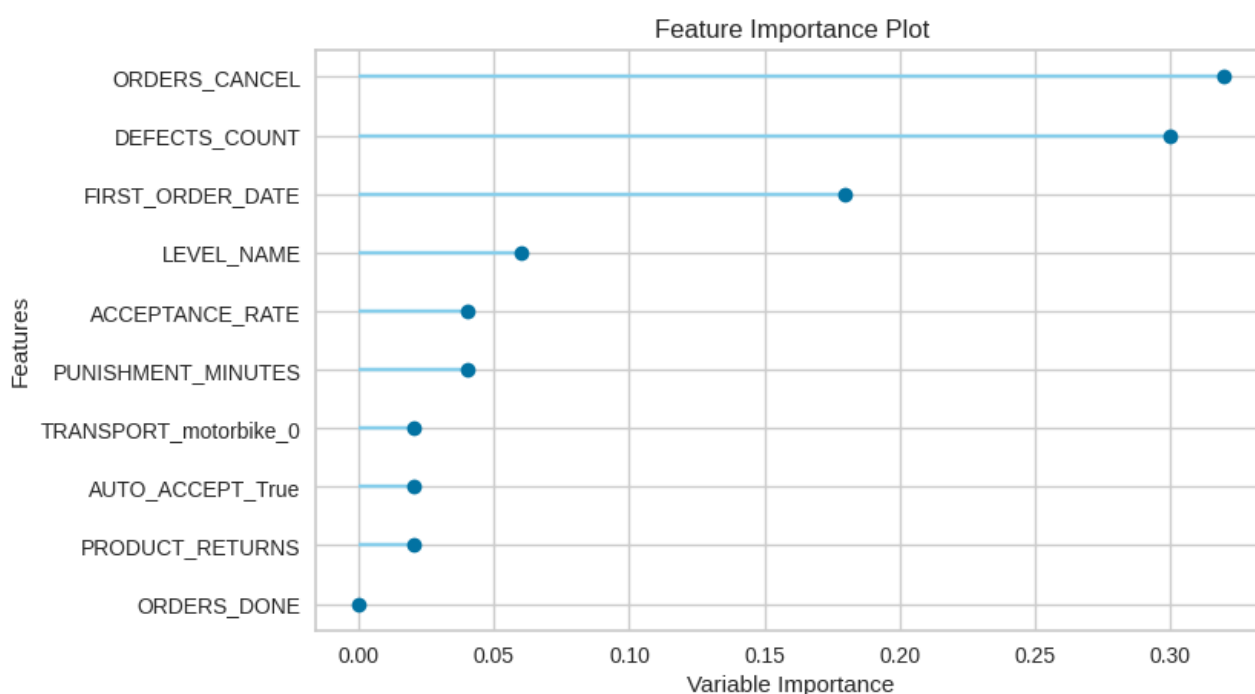
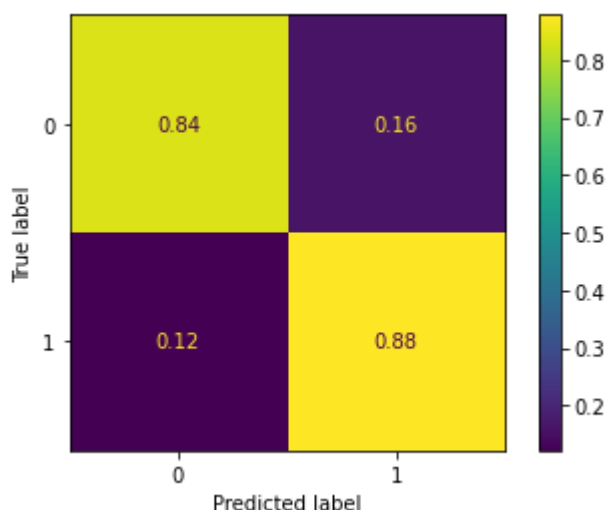


Gráfico de importância das features:



Após o ajuste dos hiperparâmetros, tivemos os seguintes resultados:

```
Acc treino: 0.9126666999650367
Acc teste: 0.8651455316659353
Revocação: 0.8780673484406324
Precisão: 0.9617811379492454
F1_score: 0.9180197389366445
```



4.5.2. Análise Comparativa

Todos os algoritmos escolhidos para avaliação foram adequados e confirmaram os resultados obtidos nos testes preliminares. Todos apresentaram acurácia acima de 85% e taxas de revocação, precisão e score F1 acima de 90%. As matrizes de confusão também apontam que os dados não estão viciados, devido a correção feita com o "undersampling".

Após o ajuste dos hiperparâmetros, obtivemos os seguintes resultados:

| Modelo | Acurácia de Treino | Acurácia de Teste | Revocação | Precisão | F1 Score |
|---------------------------------|--------------------|-------------------|-----------|----------|----------|
| Light Gradient Boosting Machine | 91,27% | 86,51% | 87,12% | 96,13% | 91,40% |
| Random Forest Classifier | 85,57% | 86,71% | 87,95% | 95,56% | 91,59% |
| Gradient Boosting Classifier | 87,49% | 86,47% | 87,16% | 96,04% | 91,38% |
| Ada Boost Classifier | 91,26% | 86,51% | 87,80% | 96,17% | 91,80% |

Notamos que o modelo Random Forest Classifier obteve a melhor acurácia quando testado e também obteve a melhor revocação. Conforme mencionado anteriormente na seção 4.4.1, esses foram os indicadores que priorizamos, portanto selecionamos o modelo **Random Forest** como o algoritmo mais adequado para solucionar o problema de predição de "churn" da Rappi.

4.6 Comparação de Modelos

5. Conclusões e Recomendações

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

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

6. Referências

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

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

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

Anexos

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