PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

NÚCLEO DE EDUCAÇÃO A DISTÂNCIA

Pós-graduação Lato Sensu em Ciência de Dados e Big Data

José Nunes da Silva Junior

ANÁLISE CHURN RATE TELECOM

Belo Horizonte

2023

José Nunes da Silva Junior

ANÁLISE DE CHURN RATE TELECOM

Trabalho de Conclusão de Curso apresentado

ao Curso de Especialização em Ciência de

Dados e Big Data como requisito parcial à

obtenção do título de especialista.

Belo Horizonte

2023

Sumário

[1 INTRODUÇÃO 4](#_Toc152015014)

[1.1 Contextualização 4](#_Toc152015015)

[1.2 O problema proposto 4](#_Toc152015016)

[2 Coleta de Dados 6](#_Toc152015017)

[3 Tratamento de Dados 10](#_Toc152015018)

[4 Analise e exploração dos dados 23](#_Toc152015019)

[5 Etapa de machine learning 48](#_Toc152015020)

[5.1 Preparando os dados para aplicação nos modelos de aprendizado de maquina 48](#_Toc152015025)

[5.2 Random Forest 53](#_Toc152015026)

[5.3 Regressão Logística 65](#_Toc152015027)

[5.4 SVM 70](#_Toc152015028)

# INTRODUÇÃO

## Contextualização

Nos últimos anos, o setor de telecomunicações tem observado um enorme avanço tecnológico, avanço este, que possibilitou novas oportunidades de negócios para as empresas. A facilidade de contratar determinados serviços com poucos cliques também é encontrada ao se tentar cancelar este mesmo serviço. Dada a crescente competição no mercado bem como o aumento da demanda dos consumidores por serviços mais personalizados e acessíveis, um elemento vem se tornando o foco das atenções: o *churn rate*.

O “*churn rate*”, ou taxa de rotatividade de clientes, é uma métrica crucial para empresas no ramo de telecomunicações, diferentemente de um modelo de compra e venda onde se obtêm o lucro no momento em que o processo se concretiza, o modelo de negócios de telecomunicações se caracterizam por um investimento inicial feito pela empresa, seja em infraestrutura, equipamentos e/ou instalação na residência do cliente. Nesse modelo de negócio, o cliente só passa a dar lucro meses ou até mesmo anos depois de contratar o serviço.

Diante disso, tornou-se fundamental para as empresas o foco nesse indicador, uma vez que a retenção de clientes é crucial para o sucesso e a sustentabilidade dos negócios.

## O problema proposto

Devido as características de investimento inicial alto e lucro futuro, a presente análise terá como foco a busca pela identificação de padrões de comportamento, além de análises demográficas, de ofertas, tempo de empresa, valor pago entre outras informações que ajudem a prever um potencial cancelamento e assim fornecer insights que ajude a empresa em suas estratégias de retenção de clientes. Na etapa de medição dos resultados dos algoritmos de aprendizado de máquina, iremos priorizar a métrica revocação (recall). A justificativa para isso é que o custo de perder clientes propensos ao churn é alto, devemos buscar minimizar o máximo possível de falsos negativos, mesmo que isso gere alguns falsos positivos.

# Coleta de Dados

O conjunto de dados *Telecom Customer Churn Prediction*, objeto de análise deste projeto pode ser encontrado através do sítio da *Maven Analytics*, uma plataforma de desafios relacionados a analises de dados.

Figura 1 - Fonte da base de dados que será utilizada.



Fonte: Maven Analytics (2022)

Na Figura 1 é possível observar os nomes e propriedades dos arquivos que serão utilizados. O conjunto em questão consiste em 3 (três) *datasets* que possuem informações e finalidades distintas (Figura 2): *telecom\_customer\_churn*, que contêm a base principal dos dados que serão analisados neste artigo; *telecom\_zipcode\_population*, que possui dados adicionais relacionados a população por código postal e; *telecom\_data\_dictionary*, que como o próprio nome sugere, trata-se de um dicionário que contém as informações do que cada coluna representa em nossa base principal.

Figura 2 - Datasets que serão utilizados



Fonte: Autoria própria.

Para uma maior facilidade de interpretação dos leitores deste artigo, os nomes das colunas, originalmente em inglês serão traduzidos para o português na etapa de tratamento de dados. Na Tabela 1 está descrito de forma detalhada as informações que cada coluna representa no arquivo *telecom\_customer\_churn*.

Tabela 1 – Detalhamento do *dataset* *telecom\_customer\_churn*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome Original** | **Nome traduzido** | **Descrição** | **Tipo** |
| CustomerID | ID\_cliente | ID única que identifica cada cliente | Texto |
| Gender | Genero | Genero do cliente | Texto |
| Age | Idade | Idade do cliente | Inteiro |
| Married | Casado | Indica se o cliente é casado | Texto |
| Number of Dependents | Qtd\_dependentes | Indica a quantidade de dependentes que moram com o cliente | Inteiro |
| City | Cidade | Cidade em que se localiza a residência do cliente | Texto |
| Zip Code | Codigo\_postal | Código postal da residência do cliente | Inteiro |
| Latitude | Latitude | Latitude da residência do cliente | Real |
| Longitude | Longitude | Longitude da residência do cliente | Real |
| Number of Referrals | Qtd\_indicações | Quantidade de indicações que o cliente realizou a amigos ou parentes | Inteiro |
| Tenure in Months | Meses\_na\_base | Indica o total de meses que o cliente permaneceu na empresa | Inteiro |
| Offer | Ultima\_oferta | Indica a última oferta que o cliente aceitou | Texto |
| Phone Service | Servico\_telefone | Indica se o cliente possui o serviço telefônico da empresa | Texto |
| Avg Monthly Long Distance Charges | Recarga\_longa\_distancia | Indica o custo médio total de recargas de longa distancia | Real |
| Multiple Lines | Multiplas\_linhas | Indica se o cliente possui mais de uma linha telefônica | Texto |
| Internet Service | Servico\_internet | Indica se o cliente possui o serviço de internet da empresa | Texto |
| Internet Type | Tipo\_internet | Indica o tipo de conexão que o cliente utiliza | Texto |
| Avg Monthly GB Download | Media\_mensal\_download\_GB | Indica o volume médio de download do cliente em gigabytes | Real |
| Online Security | Servico\_seguranca | Indica se o cliente possui um serviço adicional de segurança provido pela empresa | Texto |
| Online Backup | Servico\_backup | Indica se o cliente possui um serviço adicional de *backup* provido pela empresa | Texto |
| Device Protection Plan | Servico\_protecao\_dispositivo | Indica se o cliente possui um serviço adicional de proteção de dispositivo | Texto |
| Premium Tech Support | Servico\_suporte\_preferencial | Indica se o cliente possui um serviço adicional de atendimento preferencial | Texto |
| Streaming TV | Servico\_tv | Indica se o cliente possui algum serviço de *streaming* de TV | Texto |
| Streaming Movies | Servico\_filmes | Indica se o cliente possui algum serviço de *streaming* de filmes | Texto |
| Streaming Music | Servico\_musica | Indica se o cliente possui algum serviço de *streaming* de músicas | Texto |
| Unlimited Data | Servico\_ilimitado\_dados | Indica se o cliente possui algum serviço de dados ilimitados | Texto |
| Contract | Tipo\_contrato | Indica o tipo de contrato ao qual o cliente está fidelizado | Texto |
| Paperless Billing | Faturamento\_sem\_papel | Indica se o cliente escolheu pelo tipo de cobrança sem papel | Texto |
| Payment Method | Tipo\_pagamento | Indica o método de pagamento escolhido pelo cliente | Texto |
| Monthly Charge | Cobranca\_mensal | Indica o total mensal pago pelo cliente por todos os serviços | Real |
| Total Charges | Total\_cobranca | Indica a soma total paga pelo cliente em todo o tempo na empresa | Real |
| Total Refunds | Total\_reembolsos | Indica a soma total de reembolso recebida pelo cliente em todo o tempo na empresa | Real |
| Total Extra Data Charges | Qtd\_dados\_extras | Indica a quantidade total de dados extras contratados pelo cliente | Inteiro |
| Total Long Distance Charges | Cobrancas\_longa\_distancia | Indica a soma total paga pelo cliente em todo o tempo na empresa referente a recargas para longas distancias | Real |
| Total Revenue | Total\_cobranca\_geral | Indica a soma total paga pelo cliente de forma geral somando o valor mensal com todas as cobranças de recargas extras | Real |
| Customer Status | Situacao\_cliente | Indica o status do cliente ao final período em que os dados foram coletados: *Churned* (cancelou), *Stayed* (permaneceu cliente) e *Joined* (se tornou cliente) | Texto |
| Churn Category | Categoria\_cancelamento | Indica uma categoria que classifica o motivo de cancelamento do cliente | Texto |
| Churn Reason | Motivo\_cancelamento | Indica o motivo do cancelamento do cliente | Texto |

Fonte: Maven Analytics (2022)

O mesmo processo será realizado no arquivo *telecom\_zipcode\_population*, como pode ser observado na Tabela 2.

Tabela 2 - Detalhamento do *dataset* *telecom\_data\_dictionary*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome Original** | **Nome traduzido** | **Descrição** | **Tipo** |
| Zip Code | Codigo\_postal | Código postal em que o cliente reside | Texto |
| Population | Populacao | População total estimada do código postal | Inteiro |

Fonte: Maven Analytics (2022)

# Tratamento de Dados

Para a execução do projeto, foi utilizada a linguagem de programação interpretada Python, na sua versão 3.9.13, além do *Jupyter Notebook*, ambiente de desenvolvimento interativo baseado na Web

Nesta etapa do projeto, será utilizada majoritariamente a biblioteca *Pandas* (Figura 3) para manejo dos dados bem como de eventuais ajustes que possam ser necessários para deixar a base de dados melhor estruturada, e com isso, buscar melhores interpretações dos algoritmos de *Machine Learning*.

Figura 3 - Importação do *pandas*



Fonte: Autoria própria

Foi utilizado para a leitura dos dados, uma função própria do *pandas*, a *pd.read\_csv* (Figura 4), através dela, é possível ler um arquivo *csv* e já transforma-lo em um *dataframe* que nada mais é do que uma estrutura de dados composta por duas dimensões: linhas e colunas, similar a uma planilha.

Figura 4 - Leitura do arquivo com o pandas



Fonte: Autoria própria

Em seguida, por meio do comando *display* é possível visualizar o *dataframe* criado (Figura 5)

Figura 5 - Visualização do dataframe



Fonte: Autoria própria

Um detalhe importante sobre a função display é que ela por padrão, exibe no máximo 20 colunas, para que ela exiba todas as colunas e assim conseguirmos ter uma visão geral do nosso dataframe iremos realizar uma configuração através do comando *pd.set\_option* e alterando o parâmetro *display.max\_columns* como mostrado na Figura 6.

Figura 6 - Ajuste no máximo de colunas da função display



Fonte: Autoria própria

Dando continuidade agora realizaremos a leitura e criação do dataframe de população por código postal (Figura 7), esses dados serão agregados ao nosso dataframe principal com o objetivo de enriquecer nossa base de dados, afim de encontrar algum padrão com a quantidade de pessoas de uma determinada região, o código usado para realizar essa junção está representado na Figura 8.

Figura 7 - Criando dataframe da população por código postal



Fonte: Autoria própria

Figura 8 - Junção dos dataframes



Fonte: Autoria própria

Agora temos o nosso *dataframe* consolidado e pronto para ser realizado os devidos ajustes.

Como informado no tópico 2 deste artigo, iremos realizar o ajuste dos nomes das colunas do nosso *dataframe*, traduzindo e alterando para um padrão onde a primeira letra é maiúscula e as palavras são separadas pelo caractere *underline* (\_). O comando utilizado para realizar tal procedimento pode ser visualizado na Figura 9.

Figura 9 – Tradução dos nomes das colunas



Fonte: Autoria própria

Também será realizada a tradução dos valores contidos em todas as colunas de texto da nossa base de dados com exceção das colunas Cidade e ID\_cliente. No que diz respeito coluna Cidade, não se faz necessário a tradução desses valores por se tratarem de nomes de cidades, já a coluna ID\_cliente, possui apenas valores que identificam o cliente dentro da empresa, para nós, entretanto, trata-se de uma informação sem valor, dessa forma esta coluna será removida da nossa base de dados. Na Figura 10 está exposto o comando utilizado para remover a coluna ID\_cliente.

Figura 10 - Removendo a coluna ID\_cliente



Fonte: Autoria própria

Já na Figura 11 iremos realizar um comando para nos retornar os valores únicos contidos em cada coluna de texto, tornando mais fácil a identificação dos valores que irão necessitar ser traduzidos.

Figura 11 - Identificando os valores que precisarão ser traduzidos



Fonte: Autoria própria

A partir da identificação desses valores, podemos agora realizar a sua tradução com o comando *replace* (Figura 12).

Figura 12 - Renomeando valores da base de dados



Fonte: Autoria própria

Dando continuidade ao nosso processo de tratamento dos dados, iniciaremos agora algumas analises iniciais em cima da nossa base de dados, com o comando *df.info()* (Figura 13) iremos visualizar algumas informações sobre o nosso *dataframe*.

Figura 13 – Informações do *dataframe*



Fonte: Autoria própria

É possível observar que em nosso *dataframe*, algumas colunas possuem valores faltantes, em uma breve análise, nota-se que essas colunas são condicionadas ao valor contido em outra coluna, por exemplo, o valor presente na coluna Tipo\_internet está relacionado ao fato do cliente possuir ou não o serviço de internet (coluna Servico\_internet).

O mesmo ocorre com colunas que dependem do valor contido na coluna Servico\_telefone, que tem informações se o cliente possui ou não o serviço de telefone. Na Figura 14 foi utilizado o comando *groupby* para agrupar os dados contidos nas colunas Servico\_internet e Servico\_telefone e o comando *count()* para que o agrupamento seja feito realizando a contagem dos valores.

Figura 14 - Contagem dos valores das colunas



Fonte: Autoria própria

Se visualizarmos os valores gerados na saída do comando na Figura 14, é possível observar que a quantidade de clientes que não possuem o serviço de telefone é a mesma quantidade de clientes com valores faltantes nas colunas que detalham especificamente o serviço de telefone (Recarga\_longa\_distancia e Multiplas\_linhas), o mesmo ocorre com clientes que não possuem o serviço de internet e as colunas que estratificam esse serviço (Tipo\_internet, Media\_mensal\_download\_GB, Servico\_seguranca, Servico\_backup, Servico\_protecao\_dispositivo, Servico\_suporte\_preferencial, Servico\_tv, Servico\_filmes, Servico\_musica e Servico\_ilimitado\_dados), isso pode ser melhor observado na Figura 15, onde o comando realiza um filtro na base de dados que retorna a quantidade de valores presentes em todas as linhas onde o cliente não possui o serviço de telefone, na Figura 16, o mesmo processo é realizado nos clientes que não possuem o serviço de internet. Em ambos os casos, as colunas que detalham o serviço aparecem sem nenhum valor preenchido.

Figura 15 - Filtro de clientes sem o serviço de telefone



Fonte: Autoria própria

Figura 16 - Filtro de clientes sem o serviço de internet



Fonte: Autoria própria

A falta de campos preenchidos se torna um problema para a nossa análise dos dados bem como para a utilização de algoritmos de *machine learning* uma vez que isso pode acarretar em uma perda de contexto e, consequentemente, em uma análise incompleta e enviesada. Além disso, a presença de valores ausentes pode afetar a qualidade dos modelos de aprendizado de máquina, visto que muitos algoritmos não podem lidar diretamente com lacunas nos dados. Em algumas situações, pode ser considerada a remoção completa da linha que contém o dado faltante como uma solução, este não é o nosso caso, além de resultar numa perda quantitativa de dados, estaríamos excluindo da nossa base todos os clientes que não possuem os serviços de telefone e os serviços de internet.

Antes de escolhermos uma estratégia para solucionar nosso problema, é necessário entendermos os dados já preenchidos nessas colunas e seu contexto. Com exceção das colunas Recarga\_longa\_distancia e Media\_mensal\_download\_GB que possuem valores numéricos que informam o custo médio total de recargas de longa distância e o volume médio de download do cliente em gigabytes, respectivamente, as outras colunas possuem os valores binários “Não” e “Sim”

Para resolver esta situação estaremos atribuindo o valor “Não” nas colunas que possuem valores binários e o valor “0” nas duas colunas que possuem valores numéricos. A justificativa para essa escolha é que, uma vez que o cliente não possua o serviço de internet, por consequência ele não irá ter um serviço de filmes ou musica por exemplo, e isso pode ser replicado para as outras colunas que detalham esses dois serviços, a Figura 17 mostra o comando realizado para imputar estes dados nas respectivas colunas.

Figura 17 - atribuindo dados nos campos com valores nulos



Fonte: Autoria própria

Continuando com nossa análise, agora temos apenas duas colunas com dados ausentes, que são Categoria\_cancelamento e Motivo\_cancelamento. A própria nomenclatura dessas colunas sugere que estão relacionadas à decisão do cliente de cancelar o serviço ou não. Essa informação pode ser encontrada na coluna Situacao\_cliente. No entanto, antes de prosseguirmos com o processo de correção de dados faltantes, é necessário fazer um ajuste. Conforme indicado na Tabela 1, a coluna Situacao\_cliente contém três valores distintos: “Cancelou”, “Continua cliente” e “Se tornou cliente”

Dado que nosso principal objetivo é analisar o churn rate e tentar prever quais clientes têm maior propensão a solicitar o cancelamento, os dados relacionados aos clientes que aderiram ao serviço se tornam um elemento de interferência em nossa base de dados. Isso ocorre porque, quando se trata de análise dos cancelamentos, estamos particularmente interessados em analisar o histórico do cliente e em detectar padrões que o levaram a solicitar o termino da prestação do serviço. No caso dos recém-contratantes, simplesmente não dispomos de um período de observação suficiente para discernir esses padrões, uma vez que não tiveram tempo de estabelecer um histórico de interações significativo com o serviço.

Na Figura 18 podemos observar a quantidade de clientes por sua situação no momento que os dados foram coletados.

Figura 18 - Quantidade de clientes por situação



Fonte: Autoria própria

O comando realizado para remover estes clientes pode ser visualizado na Figura 19.

Figura 19 - Removendo clientes que contrataram o serviço no período analisado



Fonte: Autoria própria

Retornando agora para as colunas Categoria\_cancelamento e Motivo\_cancelamento, podemos realizar um agrupamento contabilizando os valores contidos nessa coluna para entender melhor os seus dados, o comando utilizando está contido na Figura 20.

Figura 20 – Agrupamento e contagem dos valores das colunas



Fonte: Autoria própria

Com base nas informações contidas no arquivo *telecom\_data\_dictionary* é possível inferir que os valores contidos na coluna Motivo\_cancelamento são uma estratificação dos valores contidos na coluna Categoria\_cancelamento, isto pode ser comprovado com a saída do comando mostrado na Figura 21, neste comando também foi realizado o agrupamento por Situacao\_cliente, comprovando que somente os clientes que cancelaram possuem dados nestas colunas.

Figura 21 - Agrupamento em conjunto das colunas



Fonte: Autoria própria

Uma vez que os valores em branco nas colunas mencionadas anteriormente são referentes aos clientes que permaneceram com o serviço, estaremos atribuindo o valor “Permaneceu cliente” em ambas as colunas (Figura 22), pois assim teremos todos os valores preenchidos sem nenhuma perca de contexto ou risco de enviesar a nossa base dados já que o cliente de fato permaneceu com o serviço.

Figura 22 - Atribuição dos valores nas linhas com dados ausentes



Fonte: Autoria própria

Finalizando esta etapa, ao executar o comando df.count() (Figura 23), sua saída nos mostra uma base de dados com 6589 registros em que todas as colunas estão devidamente ajustadas e sem valores faltantes.

Figura 23 - Contagem dos valores por coluna



Fonte: Autoria própria

# Analise e exploração dos dados

Partindo agora para a etapa de análise exploratória dos dados iremos avançar na investigação detalhada do nosso dataframe com o objetivo de identificar padrões, tendências e fatores que moldam a decisão dos clientes de permanecerem ou cancelaram seus serviços. Esta análise aprofundada permitirá a formulação de estratégias de retenção de clientes mais eficazes e aprimoramento da qualidade dos serviços oferecidos pelas empresas de telecomunicações.

Primeiramente, iremos importar a biblioteca *matplotlib*, ela será fundamental nesta etapa pois através dela conseguiremos construir gráficos que facilitarão tanto nossas analises quanto as tomadas de decisões, o comando utilizado para importar esta biblioteca pode ser visualizado na Figura 24.

Figura 24 - Importando o Matplotlib



Fonte: Autoria própria

Com a biblioteca importada, podemos iniciar a construção dos nossos gráficos, para iniciar podemos visualizar o percentual de clientes por gênero em nossa base, o código utilizado está registrado na Figura 25. Já na Figura 26 temos o gráfico gerado pelo código em questão.

Figura 25 - Código para geração do gráfico de percentual por gênero



Fonte: Autoria própria

Figura 26 - Gráfico percentual por gênero



Fonte: Autoria própria

É possível observar que possuímos uma base de dados muito equilibrada nesse aspecto. Se cruzarmos esse dado com outras colunas, podemos estratificar ainda mais essa informação, na Figura 27 por exemplo, utilizaremos um código que irá criar faixas etárias a partir da coluna “Idade”, a fim de agrupar os dados em categorias mais amplas. Esse procedimento simplificará a análise, permitindo-nos identificar padrões de maneira mais eficaz.

Figura 27 - Criação de faixas etárias



Fonte: Autoria própria

Posteriormente, exploraremos a composição de gênero dos clientes dentro de cada faixa etária, permitindo-nos visualizar essas informações de maneira mais representativa por meio de um gráfico de pirâmide. Na Figura 28 podemos visualizar o código utilizado para criação gráfico e na Figura 29 o gráfico em si.

Figura 28 - Código para criação do gráfico de pirâmide do gênero do cliente



Fonte: Autoria própria

Figura 29 - Gráfico de gênero por idade



Fonte: Autoria própria

Neste nível de estratificação conseguimos observar que as faixas etárias de 66 até 80 anos possuem uma quantidade de clientes menor em relação as outras. Iremos agora utilizar esta mesma estrutura de faixas etárias e estratificar a coluna “Situacao\_cliente”, o código para criar este novo gráfico pode ser visualizado na Figura 30.

Figura 30 - Código para criação do gráfico de pirâmide da situação do cliente



Fonte: Autoria própria

Figura 31 - Gráfico de pirâmide da situação do cliente



Fonte: Autoria própria

Na Figura 31 temos o gráfico gerado, nele é possível observar que temos uma quantidade bem menor de clientes que cancelaram em relação aos que continuam cliente, algo que já tínhamos observado anteriormente (Figura 18), porém, importante nos atentarmos que o volume referente as faixas de 66 a 80 anos diminuíram pouco em relação as outras, o que pode indicar uma maior tendência desse público em cancelar o serviço.

Para melhor visualizarmos essa hipótese, iremos realizar verificar a proporção de clientes que cancelam por faixa etária. Primeiramente iremos identificar qual o valor que corresponde a proporção geral da nossa base (Figura 32).

Figura 32 - Encontrando o valor proporcional de clientes que cancelaram



Fonte: Autoria própria

Em seguida realizaremos o mesmo procedimento para todos os valores no dataframe com as faixas etárias (Figura 33).

Figura 33 - Calculando valor proporcional por faixa etária



Fonte: Autoria própria

Com isso podemos então criar um gráfico comparando o proporcional de cancelamento de cada faixa etária com o proporcional de cancelamento geral da nossa base de dados (Figura 34).

Figura 34 - Gráfico de proporção de cancelamento por faixa etária



Fonte: Autoria própria

Através do gráfico, fica bem perceptivo o quanto os proporcionais das últimas três faixas etárias são bem mais altos em relação aos outros, chegando a valores acima de 40%. Tendo isso em mente, pode ser interessante para a empresa buscar a retenção deste perfil de cliente. Se analisarmos a retenção do cliente baseado na última oferta recebida temos o seguinte cenário (Figura 35).

Figura 35 - Retenção do cliente por oferta recebida



Fonte: Autoria própria

Uma ponderação se faz necessária nesse momento em relação aos dados da coluna “Ultimas\_ofertas”, onde, por mais que os nomes das ofertas possam sugerir isto, com as informações que temos em mãos atualmente não é possível concluir que as ofertas possuem uma relação de ordem entre si, dessa forma iremos trata-las como distintas uma da outra.

Voltando a atenção para o gráfico na Figura 35 algo que chama a atenção é a alta quantidade de clientes que cancelam o serviço sem nenhuma oferta recebida, sugerindo que esses clientes nem mesmo aceitaram negociar o cancelamento, isso por si só já é uma informação valiosa. Posteriormente iremos analisar a coluna “Motivos\_cancelamento” para entendermos melhor esse dado. Por enquanto, iremos nos aprofundar um pouco mais a retenção do cliente dado sua faixa etária e última oferta recebida.

Com o código apresentado na Figura 36 obtemos como saída uma tabela com o percentual de clientes que permaneceram na base após uma determinada oferta.

Figura 36 - Geração de tabela com a retenção por faixa etária e última oferta



Fonte: Autoria própria

Em seguida, a partir da tabela gerada é possível construir um gráfico (Figura 37) onde cada linha representa o percentual de retenção atingido por cada plano através das faixas etárias.

Figura 37 - Gráfico de retenção de cliente por faixa etária e último plano ofertado



Fonte: Autoria própria

O gráfico nos mostra que os planos têm uma efetividade diferente de acordo com a faixa etária do cliente ao qual está sendo ofertado, em alguns casos chegando próximo aos 100% de retenção. Se analisarmos as três últimas faixas etárias por exemplo, faz mais sentido tentar fidelizar o cliente com a “oferta A”, já que possui uma probabilidade de retenção do cliente maior que as outras ofertas. Um fato importante que deve ser levado em consideração é que está é uma análise puramente estatística, pois como não temos informações sobre as diferenças dos planos não é possível fazer uma análise mais profunda. É possível que a “oferta A” seja montada com serviços que atendem melhor este público de maior idade, o que explicaria essas diferenças nas retenções entre as ofertas.

Continuando com a análise da nossa tabela, temos a coluna que informa o total de meses do cliente na base da empresa, estratificando esse dado e detalhando entre clientes que cancelaram e clientes que permaneceram temos a seguinte situação (Figura 38).

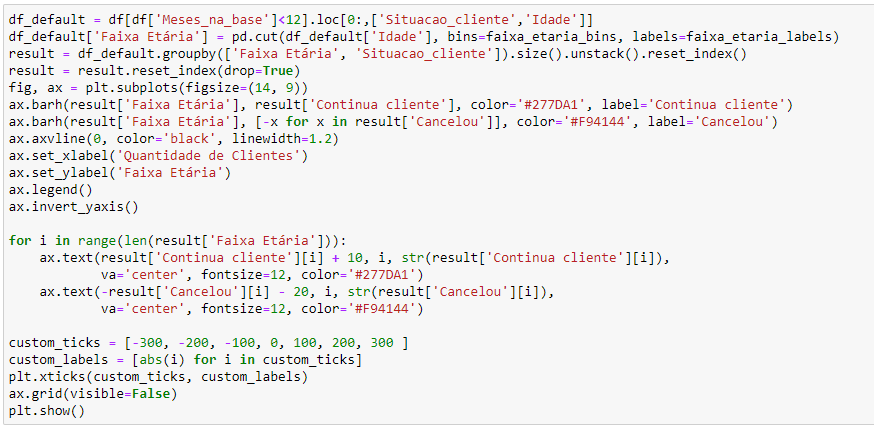
Figura 38 - Situação do cliente vs Tempo de base



Fonte: Autoria própria

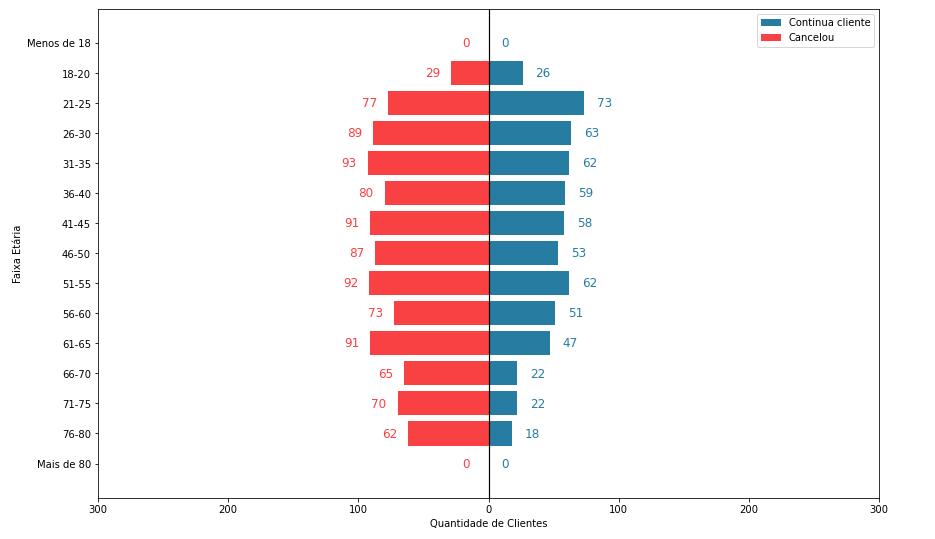
Algo bem notório no gráfico apresentado é a alta quantidade de clientes cancelando o serviço com menos de 1 ano de serviço. Estratificando esse dado pelas faixas etárias (Figura 39) temos o seguinte cenário (Figura 40).

Figura 39 - Estratificando os clientes com menos de 12 meses



Fonte: Autoria própria

Figura 40 - Situação de clientes com menos de 12 meses por faixa etária

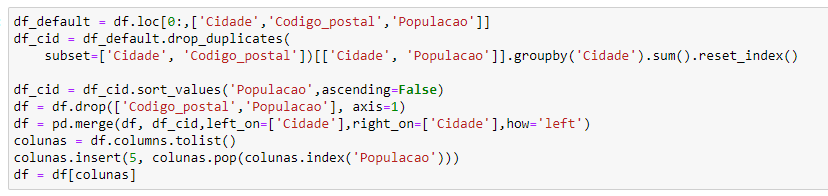


Fonte: Autoria própria

Novamente temos uma situação onde nas faixas etárias de 65 a 80 anos chama a atenção pela alta quantidade de cancelamentos em relação a quantidade de clientes que permanecem na base, comprovando que este perfil de cliente deve ser um dos focos de atenção da empresa nas suas estratégias de retenção.

Podemos aprofundar nossa análise considerando o perfil do cliente em relação à sua localidade e ao tamanho populacional dessa área. Para isso, é necessário realizar uma pequena modificação em nossa tabela. A coluna de população está atualmente vinculada à coluna de código postal, que por sua vez, possui uma relação de um para muitos com a coluna de cidades. Em outras palavras, um código postal está associado a uma única cidade, enquanto uma cidade pode abranger diversos códigos postais em sua região. Essa correção permitirá uma análise mais precisa do perfil do cliente com base na demografia da cidade em que reside. Com o ajuste realizado não será mais necessária a coluna de código postal uma vez que teremos a população por cidade. O código utilizado para realizar esse ajuste está contido na Figura 41.

Figura 41 - Ajuste dos valores da população



Fonte: Autoria própria

Em seguida, para uma melhor visualização dos dados, iremos agrupar as cidades com base na população total (Figura 42).

Figura 42 - Cidades agrupadas com base no total da população

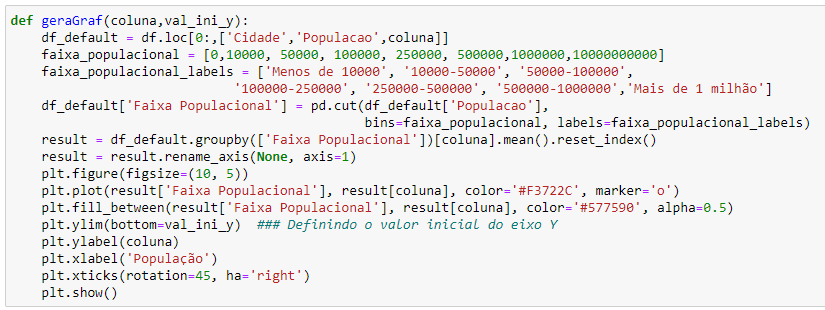


Fonte: Autoria própria

Neste momento, usaremos essa estrutura de agrupamento para a criação de alguns gráficos onde analisaremos a média de determinados valores baseados nesses grupos. Além disso, é preciso destacar que nos gráficos a seguir que serão feitos com base nesses agrupamentos, foi ajustado o valor inicial do eixo Y apenas para deixar mais visível as variações dos valores.

Para agilizar a criação dos gráficos, foi definida uma função (Figura 43) que recebera como parâmetros a coluna que irá utilizar e o valor inicial do eixo Y.

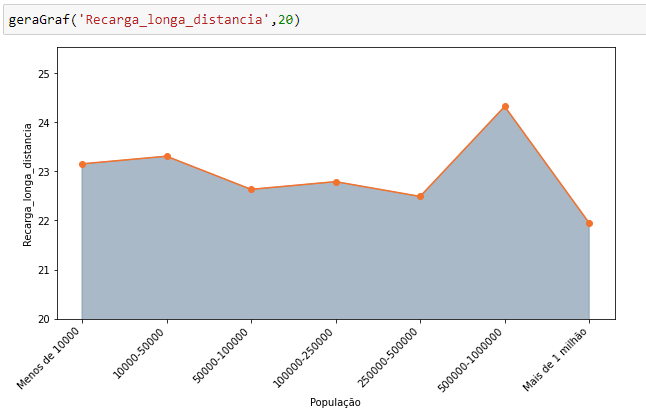
Figura 43 - Função para gerar os gráficos



Fonte: Autoria própria

Começando pela coluna de recargas de longa distância obtemos o seguinte gráfico (Figura 44):

Figura 44 - Ticket médio de recarga de longa distância

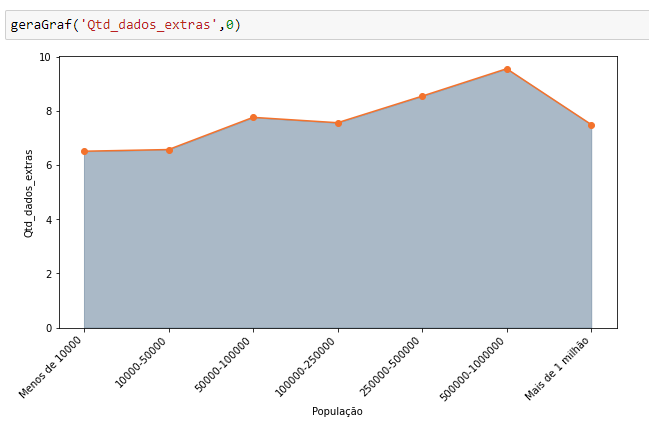


Fonte: Autoria própria

Com exceção do valor registrado no agrupamento de cidades com população entre 500000 e 1000000, o gráfico apresenta uma tendência de diminuição do ticket médio gasto em recarga de longa distância em relação ao total da população da cidade, uma possível explicação para isso pode ser que, em teoria, um cliente de uma cidade pequena pode precisar ligar para outras cidades mais do que um cliente em uma cidade grande.

Por outro lado, a solicitação para pacotes de dados extras apresenta o comportamento contrário, como pode ser visto na Figura 45.

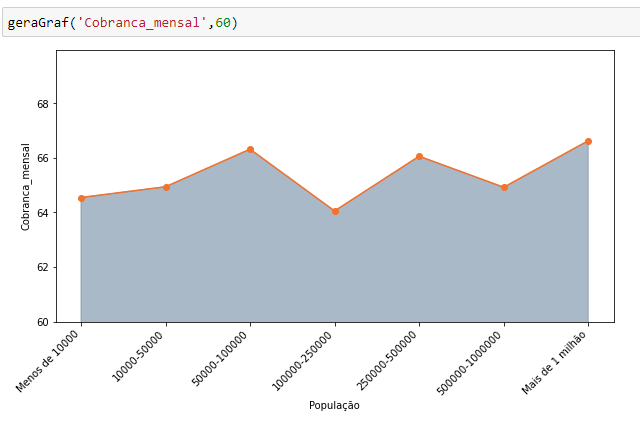
Figura 45 – Quantidade de pacotes de dados extras solicitados



Fonte: Autoria própria

Referente ao ticket médio pago pelo cliente (Figura 46) também é possível perceber uma pequena tendência de crescimento em cidades mais populosas.

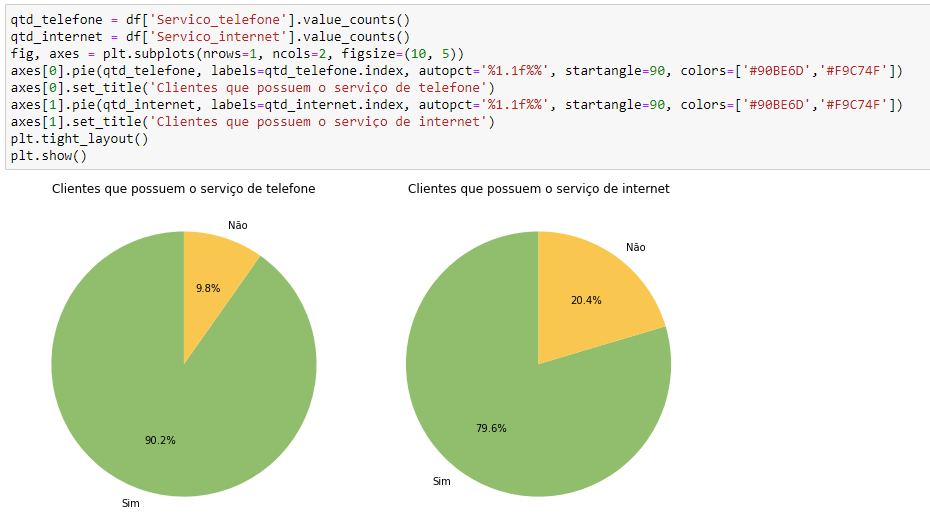
Figura 46 - Ticket médio pago pelo cliente



Fonte: Autoria própria

Partindo agora para a análise dos serviços vendidos, iremos primeiro visualizar como se dá a distribuição dos principais serviços oferecidos pela empresa, o código expresso na Figura 47 nos mostra a distribuição percentual dos serviços de telefone e internet respectivamente.

Figura 47 - Distribuição de clientes por serviço possuído



Fonte: Autoria própria

Em seguida, podemos estratificar pela população das cidades. Começando pelo serviço de telefone, na Figura 48 temos a quantidade de clientes que possuem ou não o serviço dado a população da cidade.

Figura 48 - Estratificação de clientes com o serviço de telefone por população



Fonte: Autoria própria

Em um primeiro momento nota-se uma alta quantidade de clientes em cidades com menos de 10000 habitantes, o que de fato é verdade, nos mostrando que a empresa em questão possui sua base de clientes bem distribuída em várias cidades menores, mas algo que também deve se notar é a proporção de clientes que possuem ou não o serviço, a Figura 49 elucida isso.

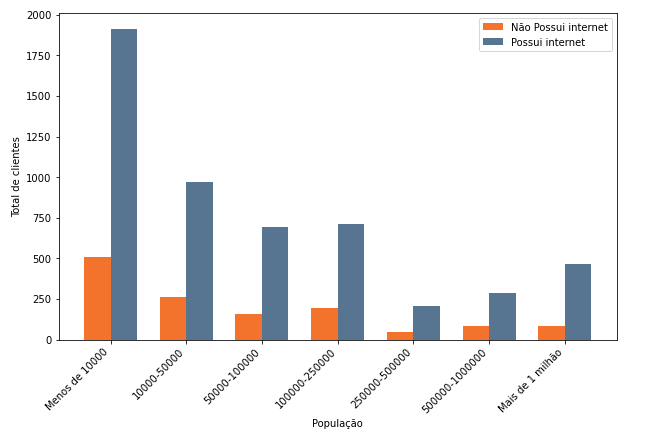
Figura 49 - Proporcional de clientes que possuem o serviço de telefone



Fonte: Autoria própria

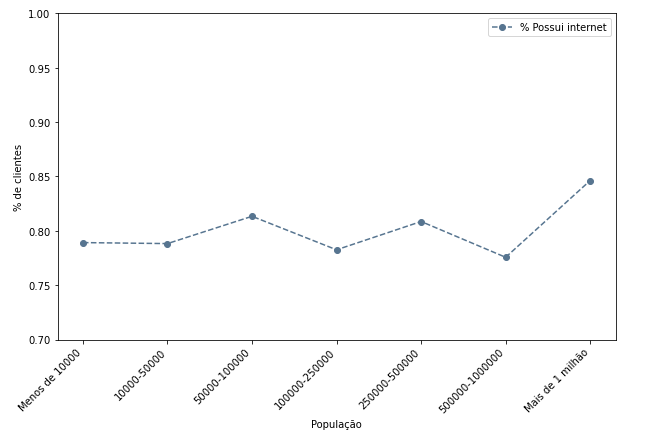
Fica nítido que as proporções se mantem quase a mesma independente da faixa populacional. O mesmo gráfico será gerado agora com a coluna referente ao serviço de internet, a Figura 50 mostra o quantitativo de clientes e a Figura 51 mostra o seu respectivo proporcional.

Figura 50 - Estratificação de clientes com o serviço de internet por população



Fonte: Autoria própria

Figura 51 - Proporcional de clientes que possuem o serviço de internet

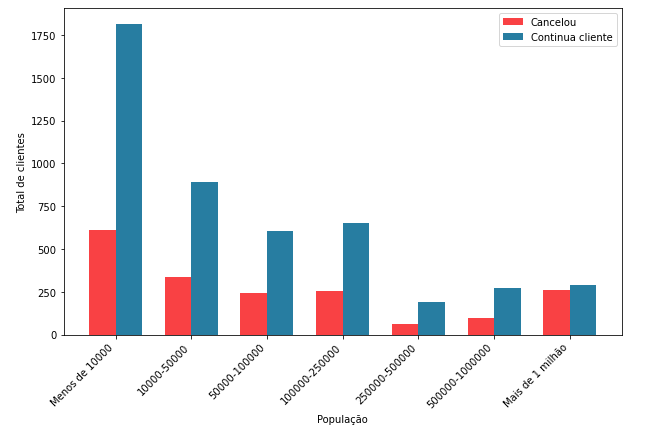


Fonte: Autoria própria

Já neste cenário, é possível observar que no último agrupamento, a taxa de clientes que possuem o serviço de internet é um pouco mais elevada em relação as demais, se encontrando próxima a 85%.

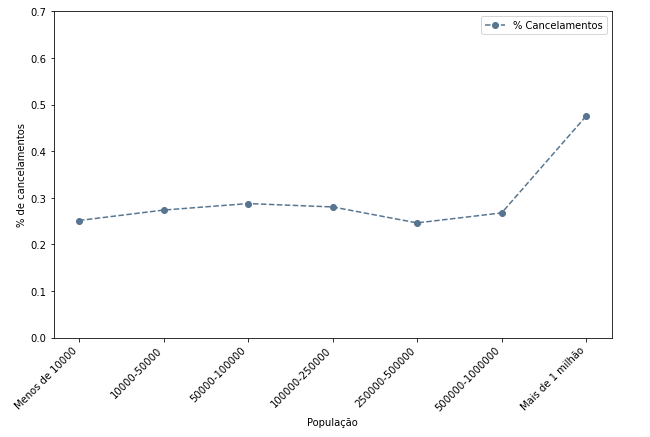
Podemos ainda, usar essa mesma estrutura para a visualização da distribuição de cancelamentos por população, na Figura 52 podemos visualizar a estratificação por quantidade de clientes e na Figura 53 o percentual proporcional de clientes que cancelaram.

Figura 52 - Estratificação da situação do cliente por população



Fonte: Autoria própria

Figura 53 - Proporcional de clientes que cancelaram



Fonte: Autoria própria

Referente ao proporcional de clientes que cancelaram, o mesmo se manteve estável com exceção do último agrupamento, onde o percentual de clientes que cancelaram sobe para aproximadamente 50%.

Outros dados que podemos explorar ainda, são os dados relacionados às vendas de serviços adicionais, que estão condicionados à posse ou à ausência de um serviço específico, que, no nosso contexto, se tratam dos serviços de internet e telefone.

Começando pelo serviço de internet, iremos primeiramente filtrar da nossa base de dados as colunas de serviços adicionais relacionadas à internet, selecionando exclusivamente os clientes que utilizam esse serviço específico. Em seguida, podemos gerar um gráfico mostrando a relação de clientes que possuem ou não o serviço adicional. Este gráfico bem como o código utilizado para sua criação pode ser visualizado na Figura 54.

Figura 54 - Distribuição de serviços adicionais de internet



Fonte: Autoria própria

Através do gráfico podemos visualizar como o serviço ilimitado de dados se destaca frente aos outros, mostrando como uma grande parte dos clientes que possuem o serviço de internet também possui este serviço adicional.

Em seguida, iremos relacionar estes dados com a retenção ou não do cliente. No código apresentado na Figura 55 estaremos, para cada serviço adicional, agrupando os dados de forma que seja contabilizado a quantidade de cancelamentos e retenções dado o fato do cliente ter ou não o serviço adicional e em seguida concatenando tudo isso em uma única tabela onde realizaremos o cálculo do percentual de clientes retidos para cada linha da tabela. Dessa forma, teremos para cada serviço, o percentual de clientes que permaneceram na empresa com o serviço adicional contratado ou não.

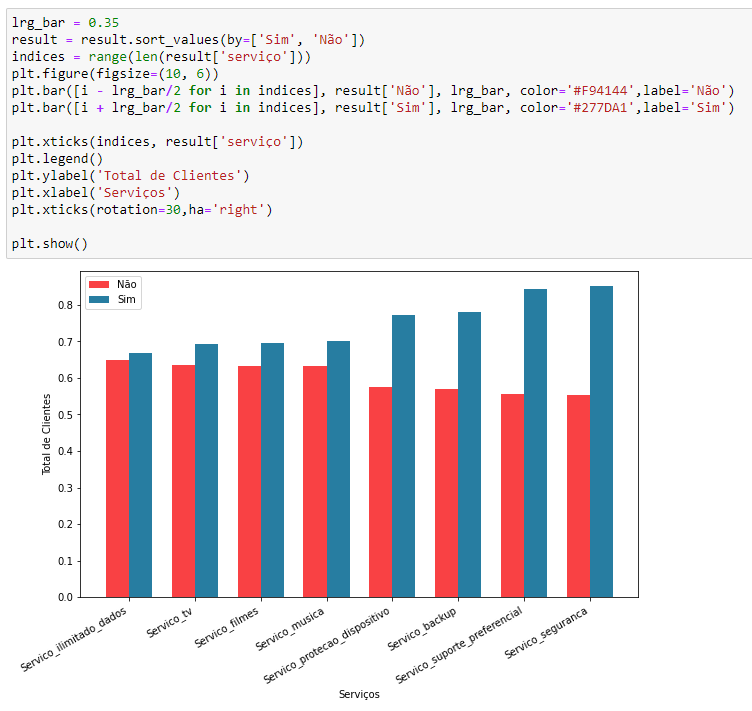
Figura 55 - Gerando percentual de clientes que permaneceram na base dado o serviço adicional



Fonte: Autoria própria

Para deixar a informação mais visual, foi utilizado o código da Figura 56 para criação de um gráfico de barras que mostra a taxa de retenção por serviço.

Figura 56 - Percentual de clientes que retidos com base na posse ou não do serviço



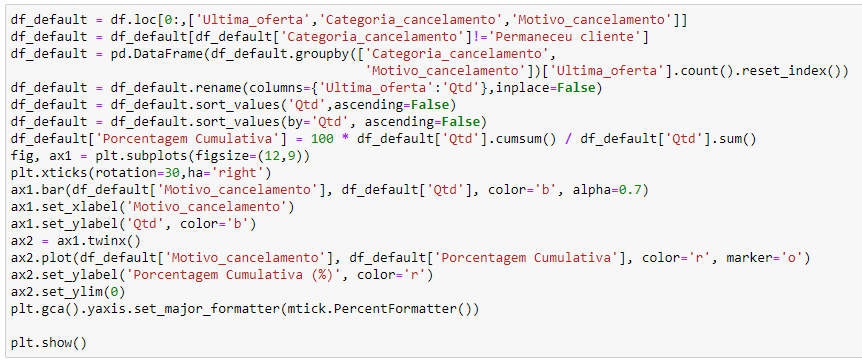
Fonte: Autoria própria

É relevante observar que, do ponto de vista estatístico, os quatro serviços vinculados à segurança e conforto apresentam uma tendência de elevar a probabilidade de retenção do cliente quando adquiridos.

Foi realizado o mesmo processo para o serviço de telefone, porém não foi encontrada nenhuma informação relevantemente expressiva.

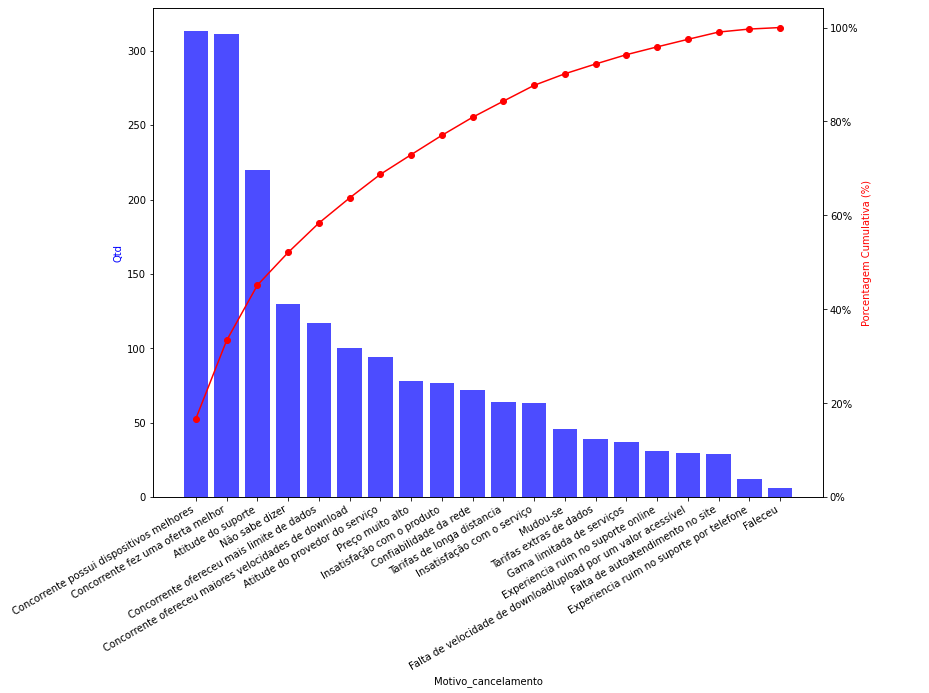
Prosseguindo, iremos examinar os motivos que levaram ao cancelamento. Inicialmente, realizaremos uma filtragem para identificar exclusivamente os clientes que optaram pelo cancelamento. Com base nesses dados selecionados, construiremos um gráfico de Pareto, proporcionando uma análise visual que destaca a frequência de cada motivo, dispostos de maneira decrescente bem como o percentual acumulado associado a cada motivo. Na Figura 57 está expresso o código utilizado e na Figura 58, o gráfico gerado.

Figura 57 - Código para criação de um gráfico de Pareto



Fonte: Autoria própria

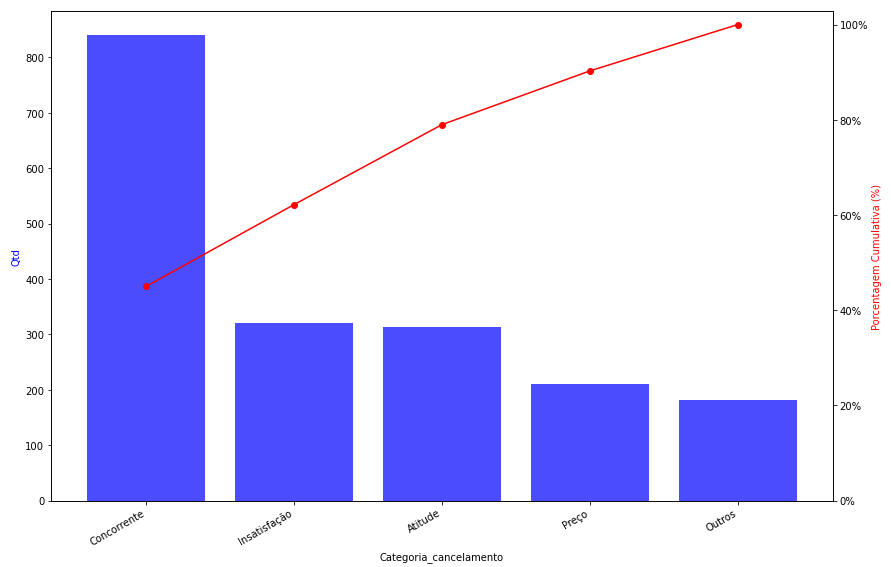
Figura 58 - Gráfico de Pareto dos motivos de cancelamento



Fonte: Autoria própria

É importante salientar que os motivos são informados pelos clientes no ato do cancelamento. Partindo do pressuposto que o cliente informou a verdade, essa pesquisa se torna uma fonte valiosa de feedback direto dos clientes sobre o que estão procurando em um produto, na situação em questão, podemos notar que 4 dos 6 motivos mais comuns estão relacionados ao concorrente. Isso pode ser melhor visualizado com o gráfico de Pareto das categorias de cancelamento realizadas pela empresa com base na informação repassada pelo cliente (Figura 59).

Figura 59 - Gráfico de Pareto das categorias de cancelamento



Fonte: Autoria própria

O princípio de Pareto nos diz que 80% dos resultados são gerados por 20% das causas. Obviamente, o principio em questão trata-se de uma heurística e não uma equação matemática formal. Dessa forma, embora a proporção específica possa variar, a ideia fundamental é que uma minoria de fatores contribui de maneira desproporcional para a obtenção dos resultados observados. Em nosso cenário, somente as causas correlacionadas ao concorrente concentram 40% dos clientes

# Etapa de machine learning

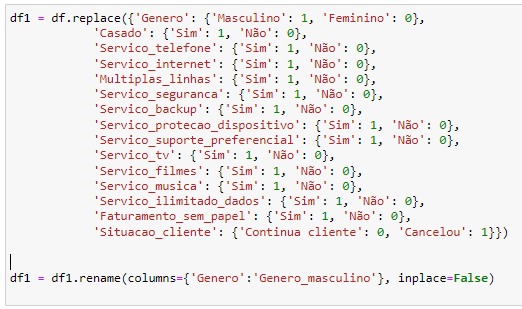


## Preparando os dados para aplicação nos modelos de aprendizado de maquina

Passaremos agora para a etapa onde utilizaremos algoritmos de aprendizado de máquina para tentar prever clientes que possam vir a solicitar o cancelamento do serviço. Para isso, alguns ajustes precisam ser feitos em nossa base.

Primeiramente, alguns algoritmos não lidam muito bem com dados em formato de texto (dados categóricos), dessa forma, precisamos alterar os valores de algumas colunas. Nas colunas que possuem valores “sim” e “não” alteraremos para 1 e 0 respectivamente. Na coluna “Situacao\_cliente” modificaremos os valores de “Continua cliente” para 0 e “Cancelou” para 1. Já na coluna de “Gênero” alteraremos o valor de “Masculino” para 1 e “Feminino” para 0, além disso, para manter consistência entre o título da coluna e a informação a ela associada, renomearemos a coluna para "Genero\_masculino", na Figura 60 podemos visualizar o código utilizado. Importante destacar que iremos realizar estes ajustes em um novo dataframe “df1” para que em etapas futuras, possamos conseguir recuperar as informações antes de serem ajustadas caso haja necessidade.

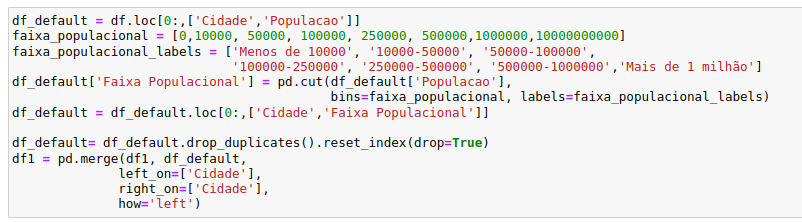
Figura 60 - Ajustando valores das colunas



Fonte: Autoria própria

Ainda em relação as colunas com valores de texto, temos a coluna de cidades que apesar de possuir valores categóricos, não podemos aplicar a mesma estratégia usada anteriormente, pois isso resultaria em atribuir a coluna, um sentido de ordem entre as cidades que na realidade não existe. Por se tratar de um dado que não podemos simplesmente remover do nosso conjunto de dados, iremos criar categorias de cidades com base em sua população assim como na Figura 42 e assim, preservaremos de uma forma mais generalizada a relação do cliente e a cidade em que reside (Figura 61). Com o agrupamento realizado, serão removidas as colunas de cidade e população.

Figura 61 - Categorização das cidades com base na população

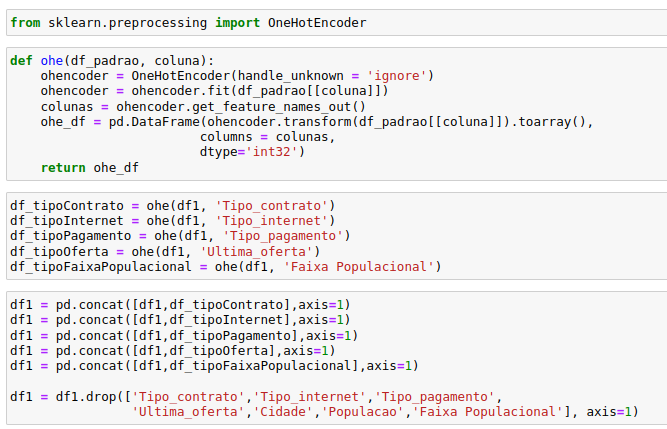


Fonte: Autoria própria

Além disso, temos algumas colunas que possuem mais de dois valores categóricos. Diferentemente da coluna cidades, essas colunas possuem quantidades pequenas de valores distintos, o que possibilita a utilização de outra estratégia. Para esses casos utilizaremos a biblioteca “OneHotEnconder”.

O que essencialmente esta biblioteca faz é converter as variáveis categóricas em representações binárias, criando uma coluna adicional para cada categoria única presente nos dados originais. Esta configuração será realizada nas colunas “Tipo\_contrato”, “Tipo\_internet”, “Tipo\_pagamento”, “Ultima\_oferta” e “Faixa Populacional” e em seguida removeremos as colunas originais uma vez que não são mais necessárias, o código referente a esta configuração pode ser visto na Figura 62.

Figura 62 - Utilização do OneHotEncoder



Fonte: Autoria própria

Em seguida foram também removidas as colunas mostradas na Figura 63, essa remoção é justificada pois as duas colunas em questão tem seus dados preenchidos apenas no caso de o cliente efetuar o cancelamento. Portanto, essas informações não podem ser utilizadas para prever o próprio cancelamento.

Figura 63 - Remoção de colunas de motivo e categoria do cancelamento

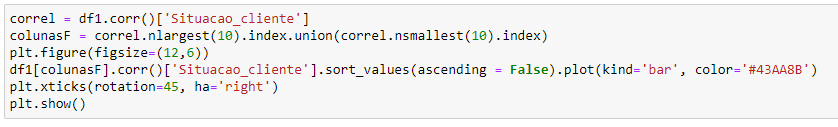


Fonte: Autoria própria

Com estes ajustes realizados, podemos gerar um gráfico para observar a correlação da coluna “Situacao\_cliente” em relação as outras.

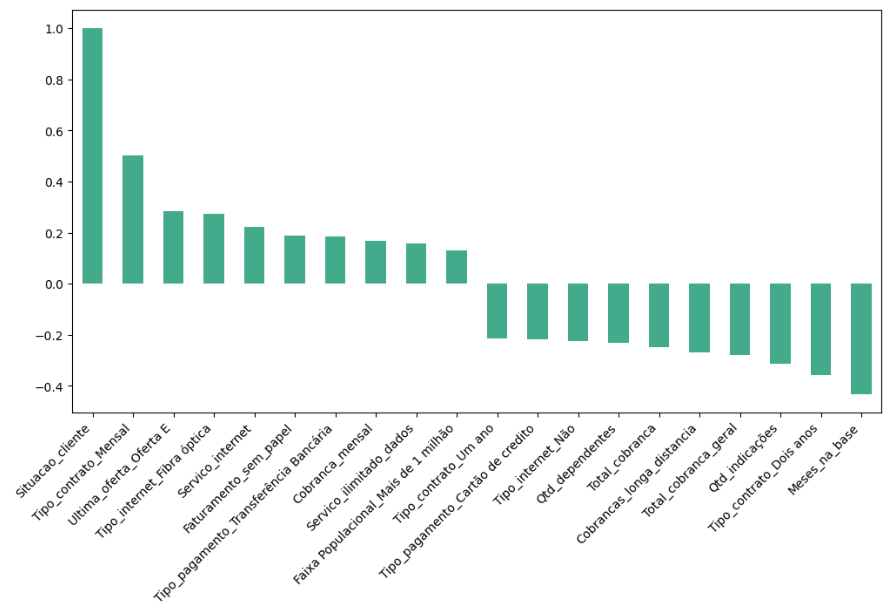
A correlação é uma medida estatística que expressa a força de um relacionamento entre duas colunas. Seu valor varia de -1 a 1, onde um coeficiente de correlação próximo a 1 indica uma forte correlação positiva, o que significa que à medida que os valores de uma coluna aumentam, os valores correspondentes na outra coluna também tendem a aumentar. Por outro lado, um coeficiente próximo a -1 indica uma forte correlação negativa, indicando que à medida que os valores de uma coluna aumentam, os valores correspondentes na outra coluna tendem a diminuir. Já um coeficiente próximo a zero sugere uma correlação fraca ou inexistente. Na Figura 64 é possível visualizar o código utilizado para gerar um gráfico com as dez colunas com maiores correlações positivas bem com as dez maiores correlações negativas, já na Figura 65 é possível ver o gráfico gerado.

Figura 64 - Código utilizado para gerar gráfico de correlação



Fonte: Autoria própria

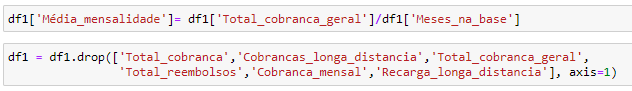
Figura 65 - Gráfico com as principais correlações da base de dados



Fonte: Autoria própria

Antes de iniciarmos a aplicação de modelos de aprendizado de máquina um último ajuste se faz necessário. Em nossa base temos algumas colunas que são redundantes, a chamada multicolinearidade, é o caso das colunas “Recarga\_longa\_distancia”, que contém a informação dos custos médios das recargas de longa distancias realizadas, e a coluna “Cobrancas\_longa\_distancia” que contem a soma total desses custos, além disso a soma desta última com a coluna “Total\_cobranca” resulta nos valores da coluna “Total\_cobranca\_geral”. Para corrigir esta situação, iremos simplificar um pouco o nosso conjunto de dados, porém sem uma perda significativa de informações. Será criada uma nova coluna que será a média dos valores pagos mensalmente pelo cliente ao longo do tempo com o serviço assinado e em seguida removeremos as colunas com informações relacionadas a custo (Figura 66).

Figura 66 - Correção de Multicolinearidade



Fonte: Autoria própria

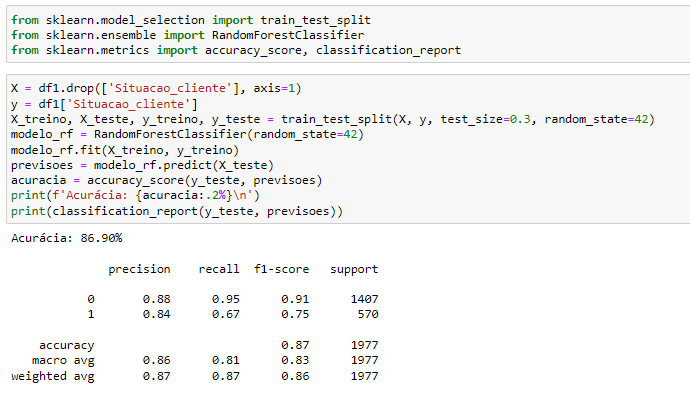
## Random Forest

Agora, iniciaremos a etapa de utilização de algoritmos de aprendizado de máquina, o primeiro a ser utilizado neste artigo será o “*Random Forest*”.

Sua escolha se dá pelo fato de se tratar de um algoritmo de aprendizado robusto, capaz de tratar base de dados complexas além de lidar bem com overfiting e ser menos sensível a ruídos.

Conforme pode ser visualizado na Figura 67, iremos importar as bibliotecas necessárias para execução do script bem como de visualização das métricas pelas quais iremos conseguir avaliar a qualidade do modelo, neste primeiro momento, após executarmos o algoritmo em suas configurações padrão, conseguimos uma acurácia de 87,15%.

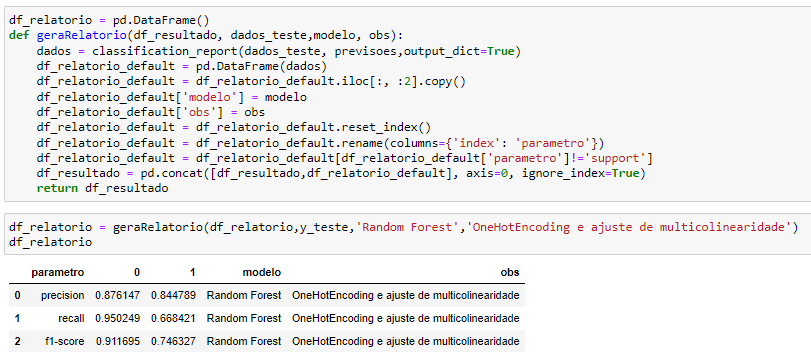
Figura 67 – Primeira execução do algoritmo Random Forest



Fonte: Autoria própria

Ainda há espaço para ajustes em nossa base de dados para conseguirmos melhores resultados nas métricas de avaliação. Tendo isso em vista, registraremos em um dataframe as métricas de precisão, revocação e f1-score de cada ajuste realizado. Assim teremos um histórico mostrando a progressão dessas métricas. A Figura 68 mostra a função criada para salvar os dados no dataframe.

Figura 68 - Função para registrar informações das métricas atingidas



Fonte: Autoria própria

Um dos ajustes que ainda precisam ser feitos, é em relação a escala dos dados. Atualmente, nossa base possui várias colunas com valores numéricos que possuem escalas de valores diferentes entre si, determinados algoritmos podem dar uma importância maior para uma determinada coluna que possui uma escala de dados maior que outra e isso acarretará na geração de resultados distorcidos. Para resolver isso, iremos realizar a normalização dos dados, ajustando cada coluna independente para que fique com a média igual a 0 e o desvio padrão igual a 1. A mostra o código que realiza tal processo.

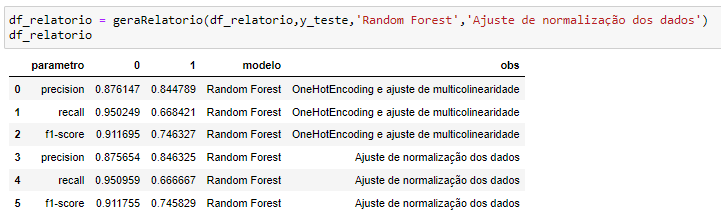
Figura 69 - Realizando a normalização dos dados



Fonte: Autoria própria

Novamente podemos executar o random forest e salvar as métricas no dataframe de relatório. Como saída, temos o resultado mostrado na Figura 70.

Figura 70 - Registrando métricas após normalização dos dados



Fonte: Autoria própria

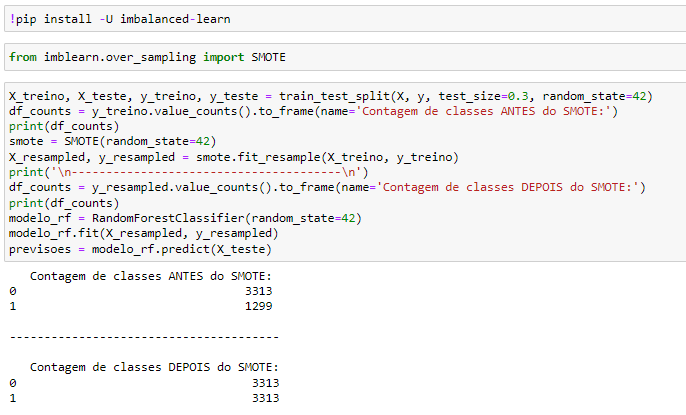
Outro fator que merece nossa atenção é o fato de nossa base de dados estar desbalanceada. Assim como já mostrado na Figura 18, temos mais de 2,5 vezes mais casos de clientes que permanecem clientes do que clientes que solicitam o cancelamento.

Duas soluções possíveis seria fazer um “undersampling” ou um “oversampling”. Na primeira opção, o excedente da classe majoritária é removido e assim iguala-se a quantidade de clientes que cancelaram com os clientes que não cancelaram. Essa solução foi descartada pois dado ao grande desbalanceamento que existe entre as classes, estaríamos perdendo muitos dados

A outra opção é justamente o contrário, utilizaremos uma técnica chamada smote que consiste em gerar amostras sintéticas, acrescentando-as dados na classe minoritária e assim igualando as quantidades das duas classes.

Se faz necessário em um primeiro momento instalar a biblioteca *imblearn* para só então conseguirmos utilizar a smote (Figura 71).

Figura 71 - Oversampling com smote



Fonte: Autoria própria

Em seguida novamente salvaremos as métricas resultantes no dataframe de relatório (Figura 72).

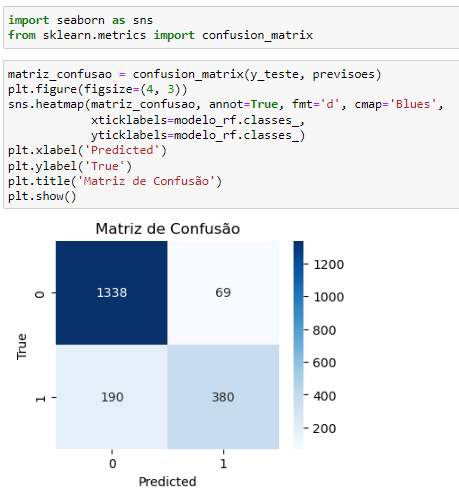
Figura 72 - Registrando métricas após balanceamento smote



Fonte: Autoria própria

Posteriormente, podemos gerar uma matriz de confusão (Figura 73) para avaliar como estão dispostas as quantidades de erros e acertos do nosso modelo.

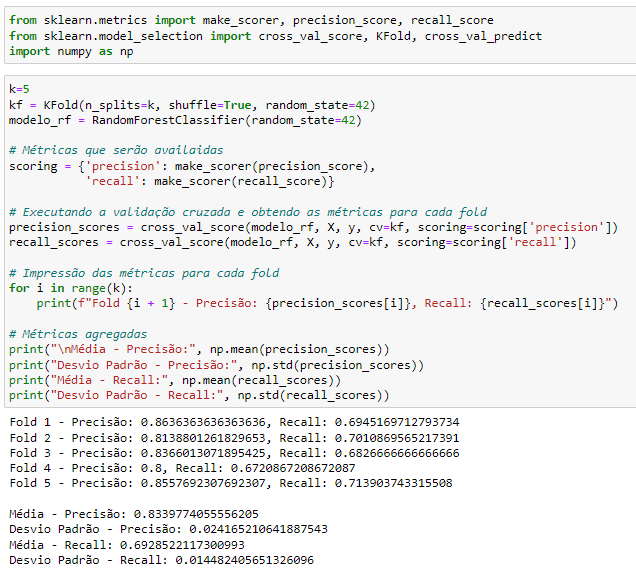
Figura 73 - Matriz de confusão



Fonte: Autoria própria

Outro fator que merece nossa atenção é em relação ao desempenho do nosso modelo e se o mesmo está apresentando overfitting, no comando mostrado na Figura 74, foi aplicado a técnica do k-fold cross validation, que consiste em dividir a base de dados em k partes (ou "folds") e, em seguida, realizar “k” iterações de treinamento e teste. Em cada iteração realizada, foi retornado na saída a precisão e a revocação e no final das iterações foi calculado também a média e desvio padrão das métricas.

Figura 74 - Analise das metricas usando kfold cross validation



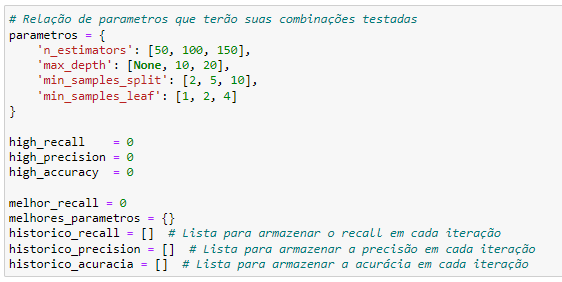
Fonte: Autoria própria

É possível observar na saída retornada no código anterior que o desvio padrão, tanto da precisão quanto da revocação são valores baixos, mostrando que o modelo se mantem bem consistente ao longo dos testes.

Como dito no inicio deste artigo, a métrica que iremos priorizar é a revocação, já que nossa preocupação é errar o mínimo possível de com cliente que irão cancelar.

Tendo isso em vista iremos prosseguir no processo de melhoria do nosso modelo e para isso iremos executar um código que percorrerá alguns laços de repetição, verificando em cada execução a combinação que retorna o melhor valor de revocação, na Figura 75 instanciamos algumas variáveis necessárias e na Figura 76 executamos os laços de repetições aninhados.

Figura 75 - Instancia das variáveis a serem usadas nos laços de repetição



Fonte: Autoria própria

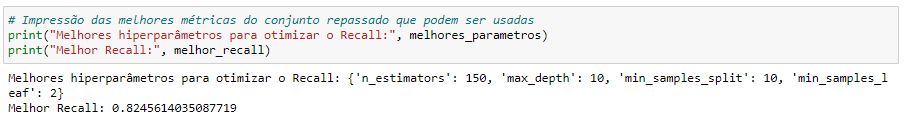
Figura 76 - Execução dos laços de repetições



Fonte: Autoria própria

Após isso iremos exibir a melhor índice de revocação encontrada bem como a combinação de parâmetros que retornou tal revocação (Figura 77).

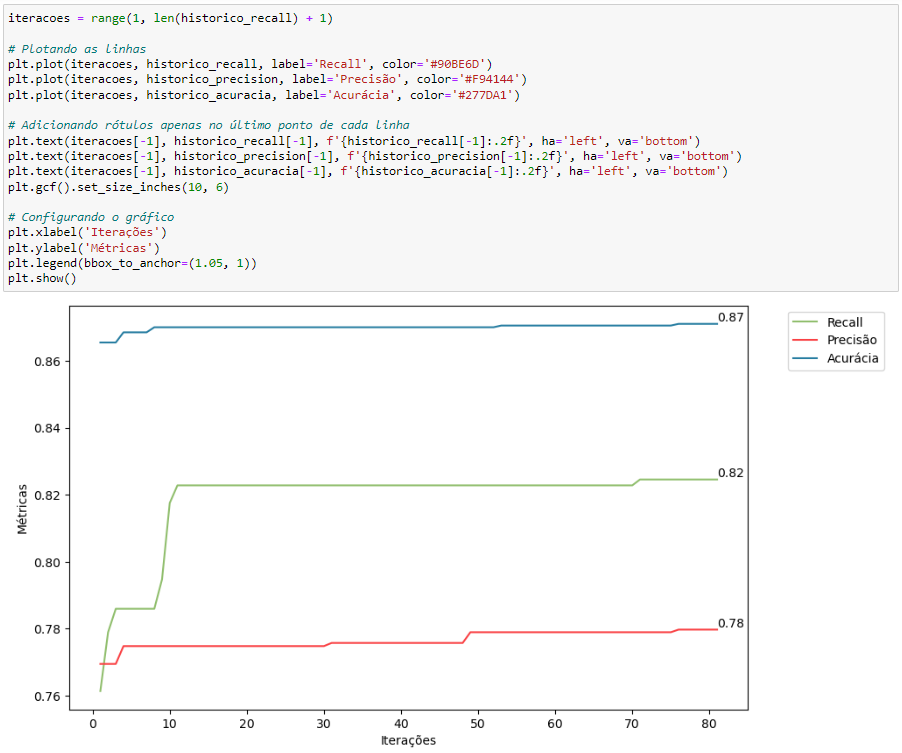
Figura 77 - Impressão dos resultados obtidos



Fonte: Autoria própria

Podemos visualizar ainda, um gráfico que mostra a evolução dos indicadores conforme os laços eram executados (Figura 78).

Figura 78 - Evolução das métricas conforme execução dos laços de repetição

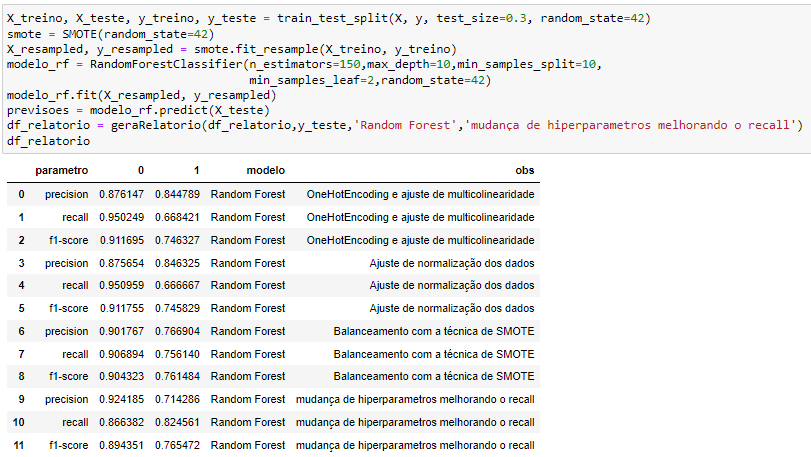


Fonte: Autoria própria

Lembrando que como o nosso foco é a revocação, utilizaremos no modelo, dentro do conjunto de parâmetros repassado, os valores que retornaram a revocação mais alta.

Executaremos o modelo mais uma vez, mas agora com os parâmetros adicionados e em seguida, salvaremos as métricas em nossa tabela de relatório (Figura 79).

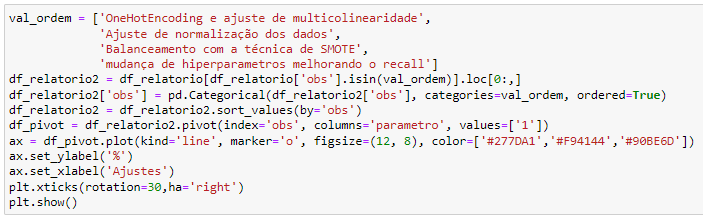
Figura 79 - Registrando métricas ajustadas na tabela de relatório



Fonte: Autoria própria

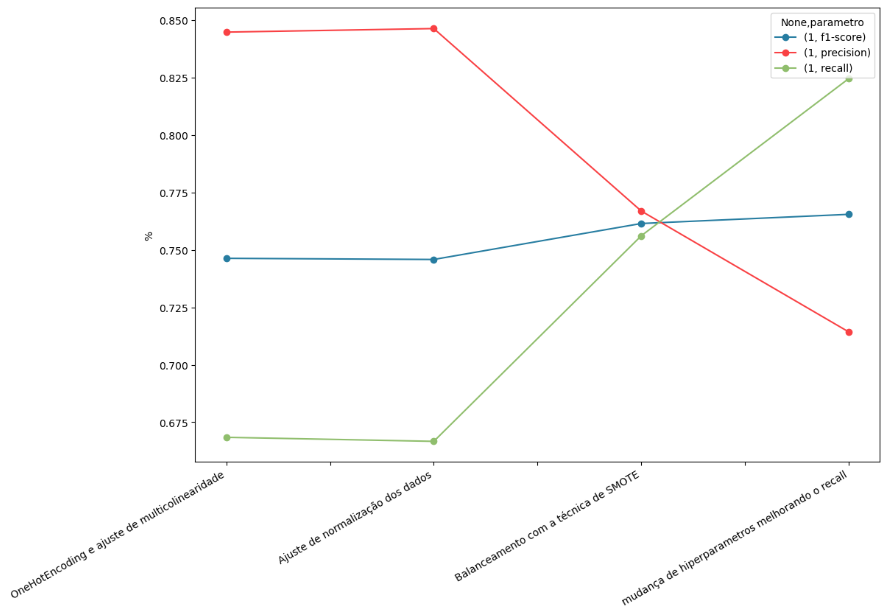
Para observarmos o efeito dos ajustes feitos até agora, iremos criar um gráfico com base na tabela de relatório em que estávamos salvando as métricas registradas. Na Figura 80 podemos ver o código utilizado e na Figura 81 o gráfico gerado.

Figura 80 - Código para geração de gráfico contendo evolução das métricas



Fonte: Autoria própria

Figura 81 - Gráfico com a evolução das métricas após cada ajuste realizado (Random Forest)

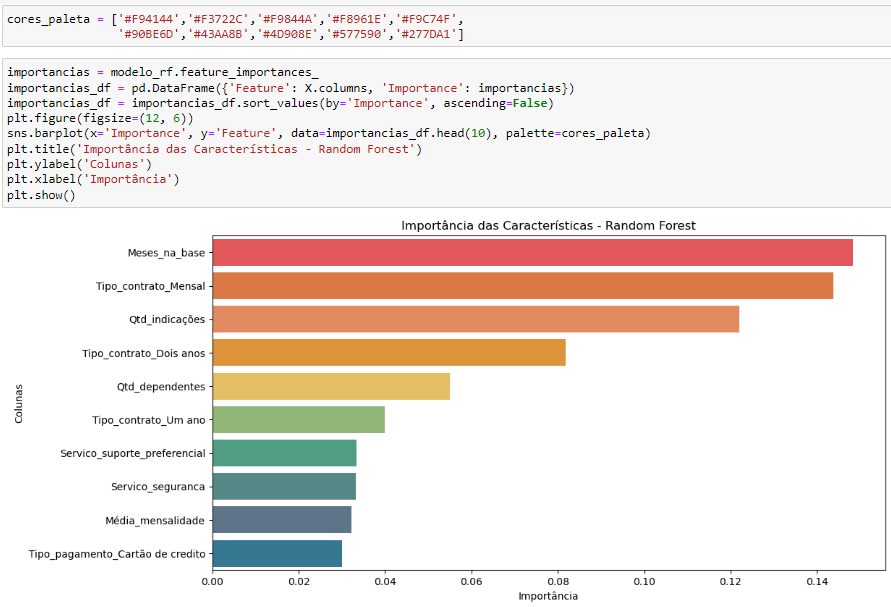


Fonte: Autoria própria

Dentre os pontos que podemos analisar, temos o fato que praticamente não houve mudança nos parâmetros após o ajuste de normalização dos dados, mostrando como o algoritmo de random forest se mostra eficiente mesmo com dados com escalas distintas. O cenário muda um pouco após realizarmos o balanceamento dos dados, mostrando que as métricas até então, poderiam estar sendo tendenciosas em relação a classe majoritária que existia na base de dados. Por fim, na última etapa, otimizamos o modelo para retornar o resultado focando em uma melhor revocação.

Outra informação interessante que podemos observar se trata das colunas que mais contribuíram para o modelo. Essas colunas e seu nível de importância podem ser observadas a partir do código executado na Figura 82.

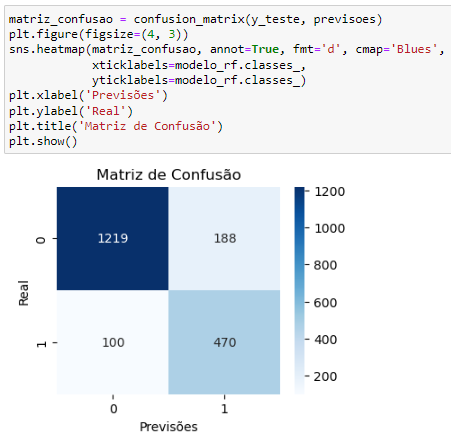
Figura 82 - Importância das colunas no modelo



Fonte: Autoria própria

Por fim, geraremos novamente a matriz de confusão para observar os dados gerados com os últimos parâmetros configurados ().

Figura 83 - Matriz de confusão com a melhor revocação



Fonte: Autoria própria

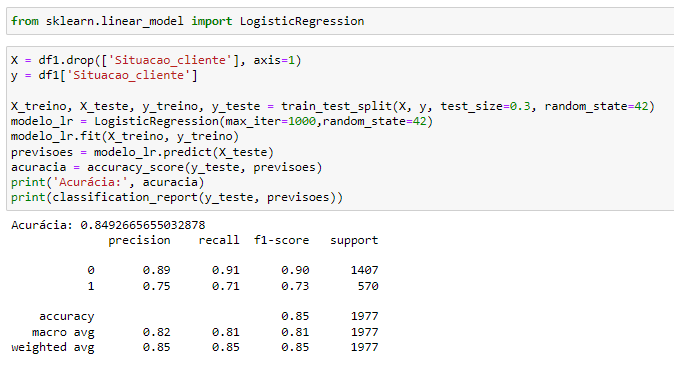
## Regressão Logística

O próximo modelo de aprendizado de máquina que iremos utilizar é o da regressão logística. Sua escolha se justifica pelo fato de possuir uma alta resistência a outliers, o que a torna mais robusta em situações onde dados incomuns possam estar presentes, além de se encaixar em nossa situação por ser um modelo de classificação binária (0 ou 1). Podemos citar ainda sua capacidade de lidar com multicolinearidade e a não necessidade de normalização dos dados. Dito isso, como tais práticas já integram nosso pipeline, optamos por realizar esses ajustes como medida preventiva, em consonância com as boas práticas de pré-processamento de dados.

Para iniciar, iremos retornar o dataframe “df1” ao estado inicial e entao repassar os passos de pré-processamento, registrando em cada etapa, as métricas de precisao, revocação e f1-scores alcançadas.

Em nossa primeira execução do algoritmo após o uso do OneHotEncoding e de ajustes de multicolinearidade conseguimos os seguintes resultados (Figura 84).

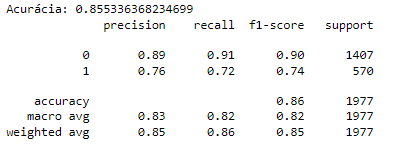
Figura 84 - Primeira execução da regressão logística



Fonte: Autoria própria

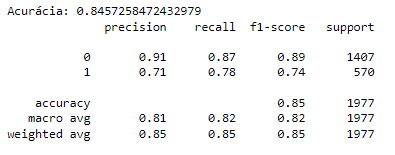
Em seguida, após realizar as etapas de normalização dos dados e balanceamento das classes obtemos respectivamente os resultados da Figura 85 e da Figura 86 respectivamente.

Figura 85 - Resultado após a normalização



Fonte: Autoria própria

Figura 86 - Resultado após o balanceamento

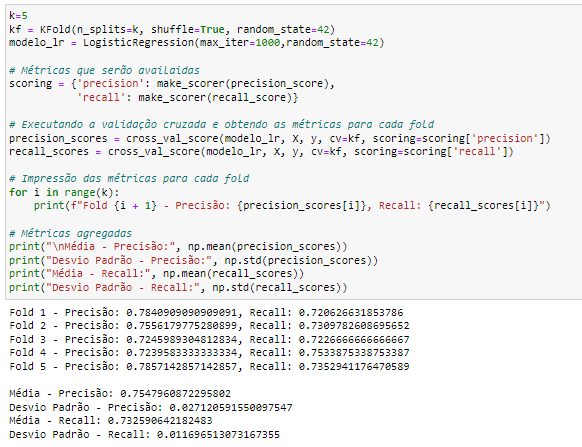


Fonte: Autoria própria

É possível observar na Figura 85 como o modelo de regressão logística lida bem com dados não normalizados, pois, mesmo após o ajuste, pouco mudou em relação as métricas de saída, diferentemente dos dados mostrados na Figura 86 que foram gerados após realizarmos o balanceamento das classes de clientes que permaneceram e que cancelaram.

Novamente iremos ainda realizar uma validação cruzada para testar o nosso modelo e, como visualizado na Figura 87, temos um modelo com um bom desempenho onde as métricas se mantem estáveis e com um desvio padrão baixo.

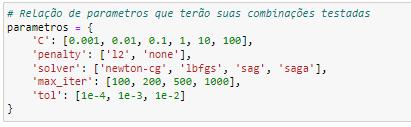
Figura 87 - Validação cruzada no modelo de regressão logística



Fonte: Autoria própria

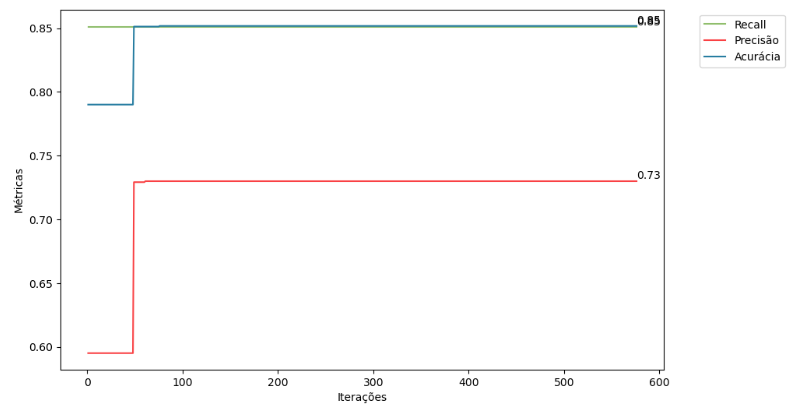
Em seguida, iremos fazer a verificação do melhor conjunto de parâmetros, na Figura 88, podemos temos os parâmetros que serão testados e na Figura 89 o gráfico mostrando a evolução das métricas conforme os laços de repetição foram sendo executados.

Figura 88 - Conjunto de parâmetros testados para a regressão logística



Fonte: Autoria própria

Figura 89 - Evolução das métricas da regressão logística

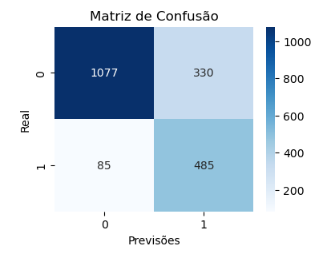


Fonte: Autoria própria

Um ponto que pode ser observado é que o valor mais alto da revocação foi atingido logo no início das execuções dos laços de repetição.

Novamente geraremos uma matriz de confusão para observarmos os quantitativos de acerto (Figura 90).

Figura 90 - Matriz de confusão da regressão logística

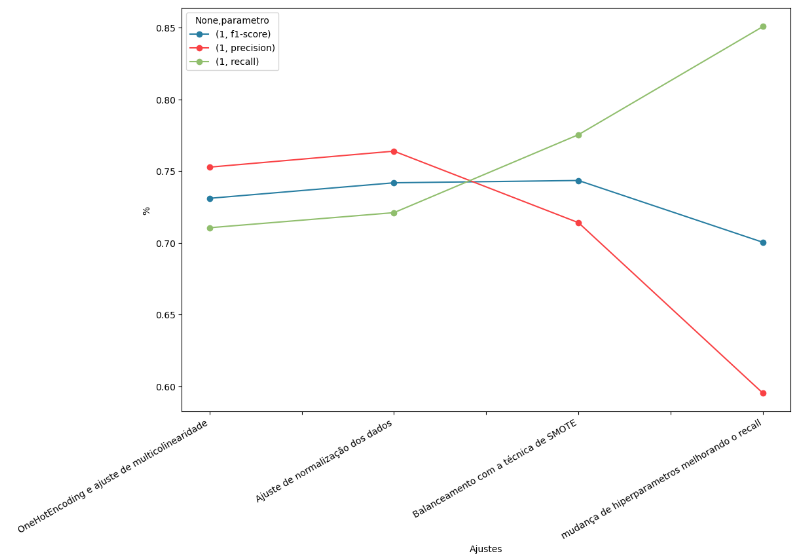


Fonte: Autoria própria

Nota-se que nossa escolha de priorizar a revocação também faz com que a precisão do modelo seja um pouco prejudicada, fazendo com que o modelo seja mais sensível ao determinar um cliente como potencial cancelamento.

A evolução dos valores das métricas após cada ajuste pode ser melhor visualizada com o gráfico da Figura 91.

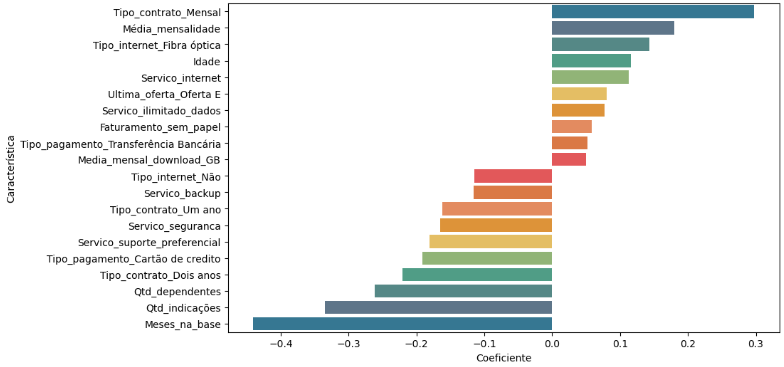
Figura 91 - Gráfico com a evolução das métricas após cada ajuste realizado (Regressão Logística)



Fonte: Autoria própria

Por fim, ao analisarmos os coeficientes do nosso modelo de regressão logística (), temos algo próximo ao resultado das colunas que mais contribuíram no modelo de random forest.

Figura 92 - Coeficientes da regressão logística



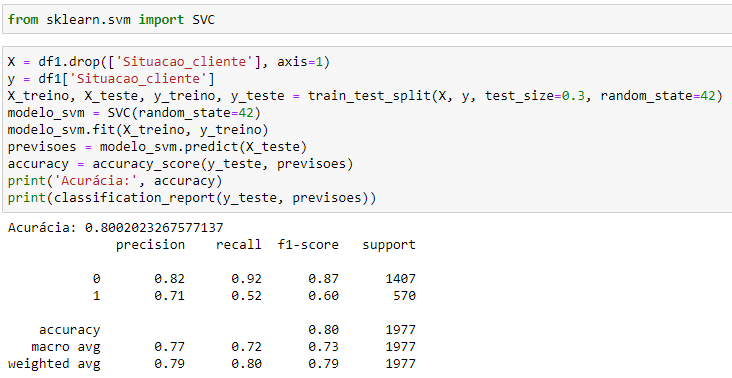
Fonte: Autoria própria

## SVM

O último algoritmo que testaremos será o SVM, sua escolha se justifica por ser um algoritmo que consegue lidar bem com base de dados com alto número de características (colunas) e ter robustez contra overfitting. Uma de suas desvantagens, porém, é sua sensibilidade ao lidar com dados com escalas muito distintas, algo que será ajustado no nosso pipeline.

Antes de tudo, iremos restaurar novamente a nossa base de dados df1 para uma fase onde temos apenas as colunas ajustadas com o OneHotEnconding e a remoção das colunas que causam multicolinearidade e em seguida, executaremos o algoritmo em suas configurações padrão (Figura 93).

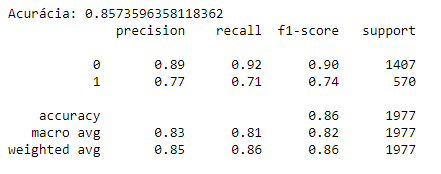
Figura 93 – Primeira execução do SVM



Fonte: Autoria própria

Ao realizarmos o ajuste da escala dos dados fica perceptível a melhoria nas métricas (Figura 94).

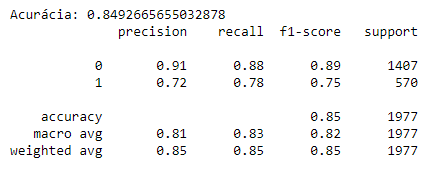
Figura 94 - Execução após o ajuste das escalas dos dados



Fonte: Autoria própria

Em seguida, após a etapa de balanceamento das classes, conseguimos obter os seguintes resultados (Figura 95).

Figura 95 - Execução após o balanceamento das classes



Fonte: Autoria própria

Ao realizarmos a validação cruzada, conforme a Figura 96, podemos observar que o modelo apresenta um bom desempenho, com um baixo valor de desvio padrão, se mostrando assim, consistente durante as execuções da validação cruzada.

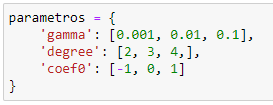
Figura 96 - Aplicação da validação cruzada no SVM



Fonte: Autoria própria

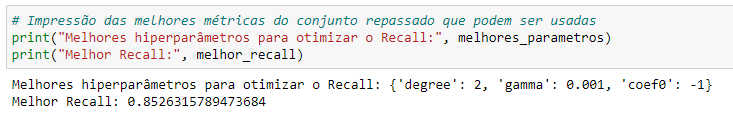
Dando prosseguimento, realizaremos agora o código com laços de repetição que irá retornar na saída o conjunto de parâmetros que resulta na melhor revocação dentre os parâmetros repassados, na Figura 97 podemos visualizar a lista de parâmetros e na Figura 98 podemos visualizar a saída resultante, com uma revocação de 85,26%.

Figura 97 - Conjunto de parâmetros testados para o SVM



Fonte: Autoria própria

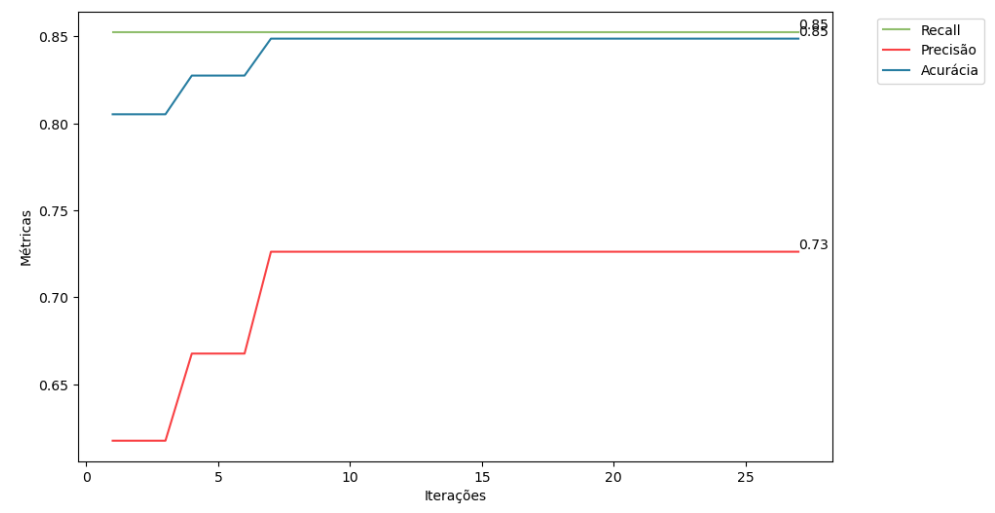
Figura 98 - Saída com o conjunto de parâmetros resultante



Fonte: Autoria própria

Na Figura 99 temos o gráfico com a evolução das métricas, nela é possível ver que a revocação mais alta foi atingida logo no início das execuções.

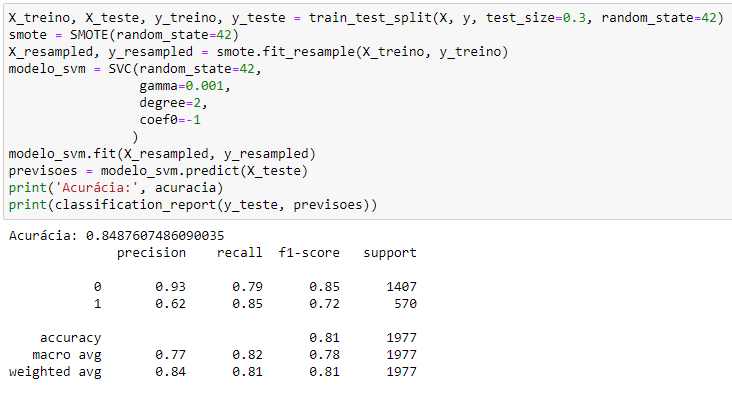
Figura 99 - Evolução das métricas do SVM



Fonte: Autoria própria

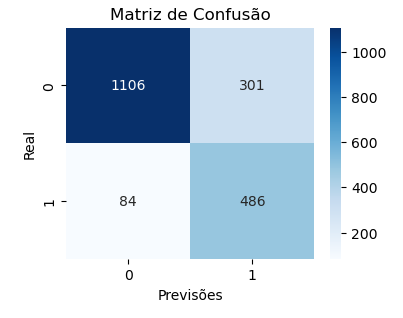
Por fim, na Figura 100 temos o resultado da execução do modelo com o conjunto de métricas encontrado e na Figura 101 geramos a matriz de confusão a fim de observamos os quantitativos de acertos do nosso modelo.

Figura 100 - Execução com o conjunto de parâmetros que foi encontrado



Fonte: Autoria própria

Figura 101 - Matriz de confusão SVM



Fonte: Autoria própria

# APRESENTAÇÃO DOS RESULTADOS

REFERENCIAS

Maven Analytics. **Telecom Customer Churn** <<https://www.mavenanalytics.io/data-playground?search=telecom>> Acesso em: 08 de out. de 2023.

Databricks. **O que é um dataframe?** <https://www.databricks.com/br/glossary/what-are-dataframes> Acesso em: 08 de out. de 2023.

Pandas. **Api Reference <**<https://pandas.pydata.org/docs/reference/index.html>> Acesso em: 08 de out. de 2023.

Entendendo o princípio de Pareto (a regra 80/20) <https://asana.com/pt/resources/pareto-principle-80-20-rule>. Acessado 17/11/2023

Introdução à correlação <https://www.aprendadatascience.com/blog/introdu%C3%A7%C3%A3o-a-correla%C3%A7%C3%A3o>. Acessado em 22/11/2023

Basta! Lidando com a multicolinearidade na análise de regressão <https://blog.minitab.com/pt/basta-lidando-com-a-multicolinearidade-na-analise-de-regressao#:~:text=A%20multicolinearidade%20ocorre%20quando%20o,certa%20forma%2C%20um%20pouco%20redundantes>. Acessado em 22/11/2023

Estudo Comparativo entre os algoritmos de Mineração de Dados Random Forest e J48 na tomada de Decisão <https://web.archive.org/web/20180424102752id_/http://revistaeletronica.unicruz.edu.br/index.php/computacao/article/viewFile/4023/737>. Acessado em 23/11/2023