UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
INSTITUTO METRÓPOLE DIGITAL
CURSO DE TECNOLOGIA DA INFORMAÇÃO

PROJETO DA SEGUNDA UNIDADE: TELCO CUSTOMER CHURN

TURMA: IMD3002 - APRENDIZADO DE MÁQUINA SUPERVISIONADO

GRUPO:
ANA LUIZA MEDEIROS DA SILVA
MARIANA TIMBÓ DE OLIVEIRA

Natal, RN
03/06/2025

1 - INTRODUÇÃO

Em uma empresa de telecomunicações, compreender o perfil dos clientes é essencial para a tomada de decisões estratégicas. A partir das informações contidas na base de dados disponível, que reúne dados de clientes de uma empresa, torna-se viável realizar um estudo utilizando técnicas de aprendizado de máquina supervisionado com o objetivo de prever se um cliente irá cancelar ou não o seu plano. Por meio de uma preparação dos dados, implementação de algoritmos de classificação e avaliação dos modelos utilizados, é possível fazer análises relevantes ao problema, contribuindo para a tomada de decisões mais assertivas.

2 - METODOLOGIA

Após compreensão preliminar dos dados, foram realizadas uma série de etapas para organizá-los e aplicá-los em modelos de aprendizado de máquina.

2.1 - Exploração Inicial dos Dados

Inicialmente, realizou-se uma etapa de exploração dos dados por meio do método .info(), com o intuito de obter uma visão geral do DataFrame e identificar a presença de valores ausentes. Foi verificado que não havia dados faltantes no conjunto. Em seguida, utilizou-se o método .duplicated().any() para identificar a presença de registros duplicados, e constatou-se que não havia duplicações. Posteriormente, com o uso do método .dtypes, foram inspecionados os tipos de dados de cada coluna. Observou-se que a coluna "TotalCharges", embora represente valores numéricos, estava classificada como *object*. Considerando sua natureza quantitativa, o ideal seria que estivesse categorizada como *float*, para permitir análises e processamentos numéricos adequados.

2.2 - Tratamento de Dados Inconsistentes

Após a identificação da coluna "TotalCharges", foi utilizado um código para converter todos os seus valores para o tipo *float*. Em seguida, aplicou-se um comando para exibir os valores que não puderam ser convertidos, totalizando 11 casos de falha na conversão. Diante disso, foi realizada uma nova tentativa de conversão que, caso não fosse bem-sucedida, já removia automaticamente os valores inválidos. Após esse processo, utilizou-se o método .dtypes para verificar o tipo da coluna, confirmando que havia sido convertida corretamente para *float*. Como resultado, passaram a existir 11 dados faltantes. Considerando o tamanho do conjunto de dados, essa quantidade foi considerada irrelevante, optando-se por excluir esses registros.

2.3 - Modificação das Colunas Categóricas para Numéricas

Com o objetivo de facilitar a aplicação de algoritmos de aprendizado de máquina, foi realizada uma análise das variáveis únicas de cada coluna, a fim de identificar padrões que, inicialmente, não eram evidentes. Durante essa etapa, observou-se uma repetição significativa do valor categórico "No internet service" nas colunas *OnlineSecurity*, *OnlineBackup*, *DeviceProtection*, *TechSupport*, *StreamingTV* e *StreamingMovies*, todas com a mesma quantidade de ocorrências. Diante disso, decidiu-se tratar esse valor como equivalente a "No", considerando que a ausência de serviço implica na ausência da funcionalidade em

questão. Assim, os valores dessas colunas foram padronizados para “Yes” e “No”, e em seguida transformados em valores binários, sendo *Yes = 1* e *No = 0*. Para facilitar esse processo, foi criada uma nova coluna chamada “HasInternetService”, e os valores “No internet service” foram substituídos por “No” nas colunas mencionadas. A mesma lógica foi aplicada à coluna PhoneService e às colunas relacionadas, realizando a substituição de “No phone service” por “No” e após essa padronização, os dados foram convertidos para o formato binário. Além disso, a coluna *Gender* também foi transformada em uma variável binária, sendo “Female” representado por *1* e “Male” por *0*. Após essas modificações, a coluna “HasInternetService” foi removida por representar uma informação redundante. Como as colunas relacionadas já indicam a ausência de serviço quando o cliente não possui internet, mantê-la poderia reforçar um padrão óbvio e contribuir para o overfitting do modelo.

2.4 - One-hot-encoding e Modificação de Colunas

Para tratar os atributos categóricos restantes, aplicou-se a técnica de *one-hot encoding* por meio de métodos da biblioteca pandas. Essa abordagem foi escolhida porque as colunas categóricas possuíam, em média, apenas três valores únicos, o que evita um aumento excessivo da dimensionalidade do conjunto de dados. A coluna “InternetService_No” foi removida por ser redundante após a codificação. Em seguida, as colunas com valores booleanos foram convertidas para o tipo *int*. Para finalizar as preparações para aprendizado de máquina o index do DataFrame foi redefinido para “customer_ID”, uma vez que esse atributo não é relevante para o aprendizado do modelo. Por fim, a ordem das colunas foi reorganizada: as colunas “MonthlyCharges” e “TotalCharges” foram posicionadas após o tipo de pagamento, e a coluna de label foi movida para o final.

2.5 - Análise do Banco de Dados

Primeiramente, a análise foi realizada com colunas de valores numéricos: “Tenure”, “MonthlyCharges” e “TotalCharges”. Utilizaram-se gráficos de boxplot e histogramas com curva de densidade, e nenhum outlier foi identificado. A variável “MonthlyCharges” apresentou maior concentração em torno do valor 20. “TotalCharges” concentrou-se em valores mais baixos, enquanto “Tenure” teve maior frequência nos primeiros meses e também em períodos superiores a 70 meses. Em seguida, foi realizada a contagem de valores por categoria, revelando que o número de homens e mulheres é semelhante. A maioria dos clientes não é idoso, não possui dependentes, utiliza serviço telefônico e não possui segurança online, entre outras observações. Além disso, é importante salientar que as classes

do banco de dados são desbalanceadas, sendo 5163 objetos para não churn e 1869 para churn, o que irá influenciar diretamente no aprendizado de máquina.

2.6 - Implementação de Algoritmos de Classificação

Para aplicar os algoritmos de classificação, optou-se pela divisão dos dados em conjuntos de treino e teste. Após essa divisão, foi realizada a normalização dos atributos numéricos “tenure”, “MonthlyCharges” e “TotalCharges”, com o objetivo de melhorar o desempenho dos modelos. A normalização foi feita somente após a divisão, a fim de evitar vazamento de dados entre os conjuntos. Foi utilizada a normalização por MinMaxScaler(), da biblioteca sklearn.preprocessing. Esse tipo de normalização se adequa melhor a dados que já possuem uma distribuição aproximada e sem outliers extremos, que é o caso do conjunto de dados trabalhado. Após a divisão e normalização dos dados, foi aplicado os seguintes modelos de classificação:

1. “SVC”: Esse algoritmo funciona buscando uma linha (ou hiperplano) que melhor separa as classes, mantendo a maior margem possível entre os grupos. O hiperparâmetro que mais influenciou no desempenho foi o kernel = “linear”, pois os dados apresentaram uma separação que pôde ser feita de forma simples, sem a necessidade de transformar os dados para outras dimensões.
2. Regressão Logística: Esse modelo é um algoritmo de classificação usado para prever a probabilidade de uma observação pertencer a uma determinada classe. Nesse caso, foram usados os hiperparâmetros padrão, como o solver 'lbfgs' (otimizador).
3. “Random Forest”: Esse modelo é um conjunto de várias árvores de decisão que trabalham juntas para melhorar a precisão da classificação. Nesse caso, foram escolhidos os hiperparâmetros n_estimators=100, que define o número de árvores na floresta, e max_depth=10, que limita a profundidade máxima de cada árvore para evitar que elas se tornem muito complexas e se ajustem demais aos dados de treino.

2.7 - Ferramentas e Bibliotecas Utilizadas

Python, Pandas, Numpy, Matplotlib, Seaborn, Gdown , Jupyter Notebook, Sklearn e suas bibliotecas.

3 - RESULTADOS

Após a implementação de algoritmos de classificação foram obtidos resultados com um nível de acurácia abaixo do esperado. Os modelos de SVC e Regressão Logística apresentaram acurácia global de 0.80 e o modelo “Random Forest” apresentou acurácia global de 0.79.

	precision	recall	f1-score	support
0	0.85	0.89	0.87	1033
1	0.64	0.55	0.59	374
accuracy			0.80	1407
macro avg	0.74	0.72	0.73	1407
weighted avg	0.79	0.80	0.79	1407

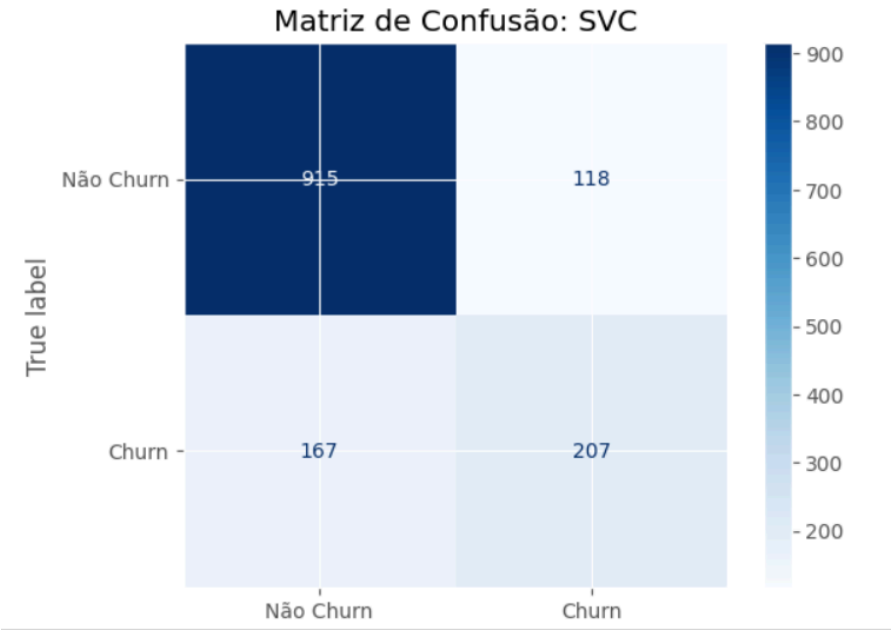


Imagem 1: Matriz de Confusão e Métricas de Avaliação do Método SVC.

	precision	recall	f1-score	support
0	0.85	0.89	0.87	1033
1	0.65	0.57	0.61	374
accuracy			0.80	1407
macro avg	0.75	0.73	0.74	1407
weighted avg	0.80	0.80	0.80	1407

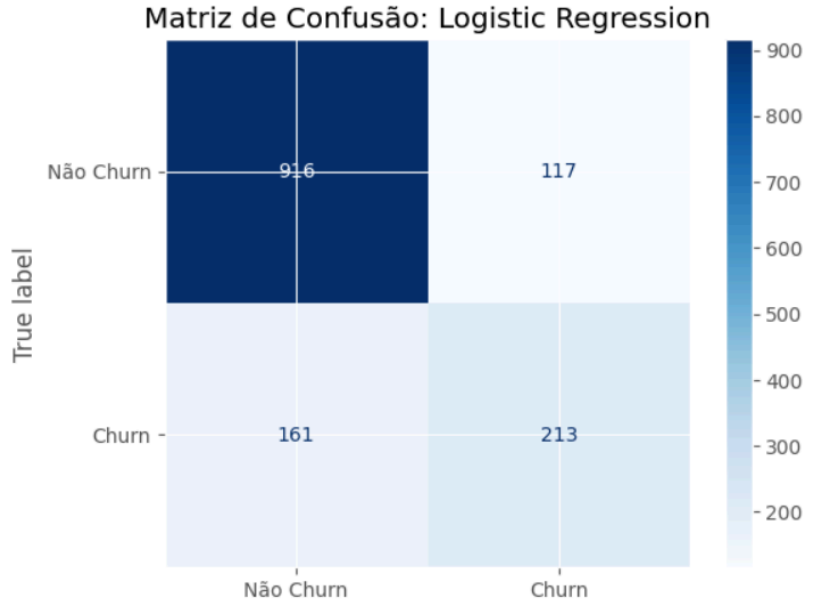


Imagem 2: Matriz de Confusão e Métricas de Avaliação da Regressão Logística.

	precision	recall	f1-score	support
0	0.84	0.89	0.86	1033
1	0.63	0.52	0.57	374
accuracy			0.79	1407
macro avg	0.74	0.71	0.72	1407
weighted avg	0.78	0.79	0.79	1407

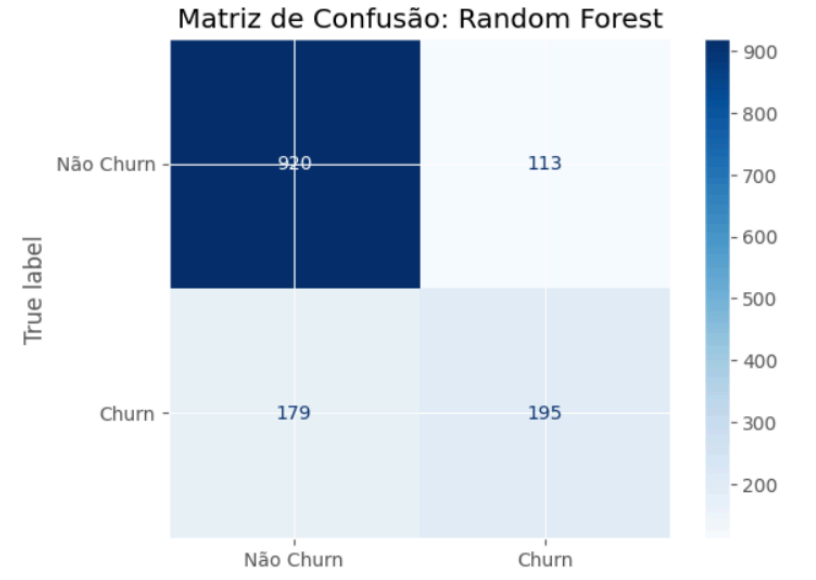


Imagem 3: Matriz de Confusão e Métricas de Avaliação do Método “Random Forest”.

Ao analisar todos os resultados do modelo, foi percebido que melhor aprendizado na classe majoritária não churn. Assim, visando buscar um melhor desempenho, testamos técnicas de balanceamento como oversampling e undersampling. O oversampling aumenta os registros da classe minoritária, copiando ou criando novos exemplos para equilibrar o conjunto e ajudar o modelo a aprender melhor essa classe. Já o undersampling diminui os registros da classe majoritária, removendo alguns exemplos para deixar as classes mais equilibradas. Após aplicar esse pré processamento, é feito novamente o aprendizado pelos modelos. Veja os resultados:

	precision	recall	f1-score	support
0	0.89	0.76	0.82	1033
1	0.52	0.74	0.61	374
accuracy			0.75	1407
macro avg	0.71	0.75	0.72	1407
weighted avg	0.79	0.75	0.76	1407

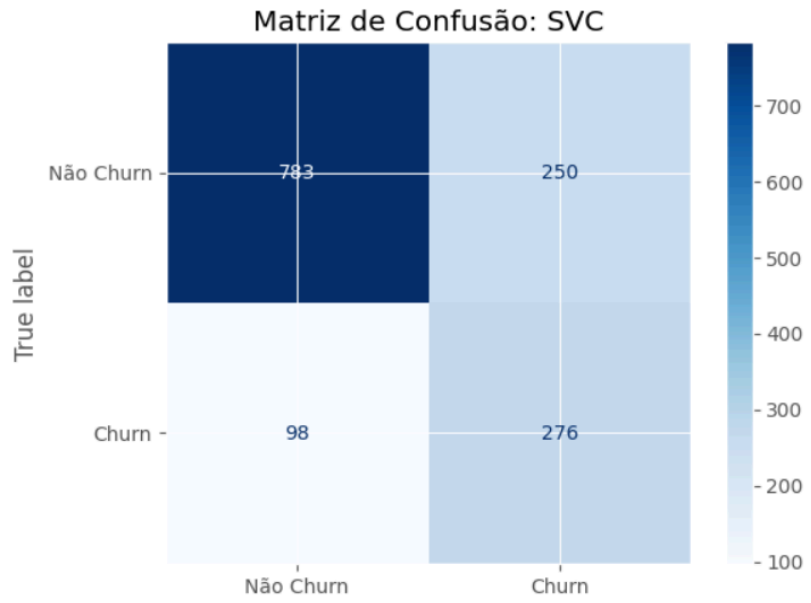


Imagem 4: Matriz de Confusão e Métricas de Avaliação do Método “SVC” com Oversampling.

	precision	recall	f1-score	support
0	0.89	0.75	0.81	1033
1	0.52	0.75	0.61	374
accuracy			0.75	1407
macro avg	0.71	0.75	0.71	1407
weighted avg	0.79	0.75	0.76	1407

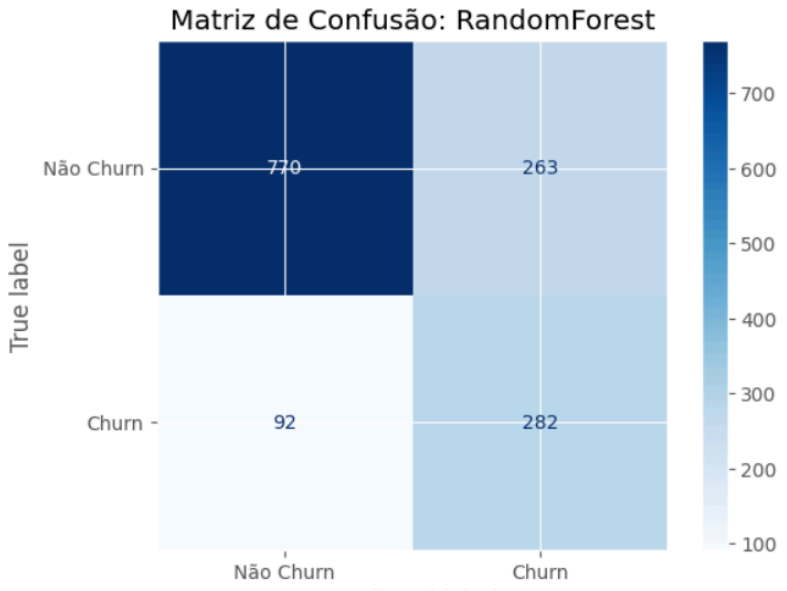


Imagem 5: Matriz de Confusão e Métricas de Avaliação do Método “Random Forest” com Oversampling.

	precision	recall	f1-score	support
0	0.91	0.64	0.75	1033
1	0.45	0.82	0.58	374
accuracy			0.69	1407
macro avg	0.68	0.73	0.67	1407
weighted avg	0.78	0.69	0.70	1407

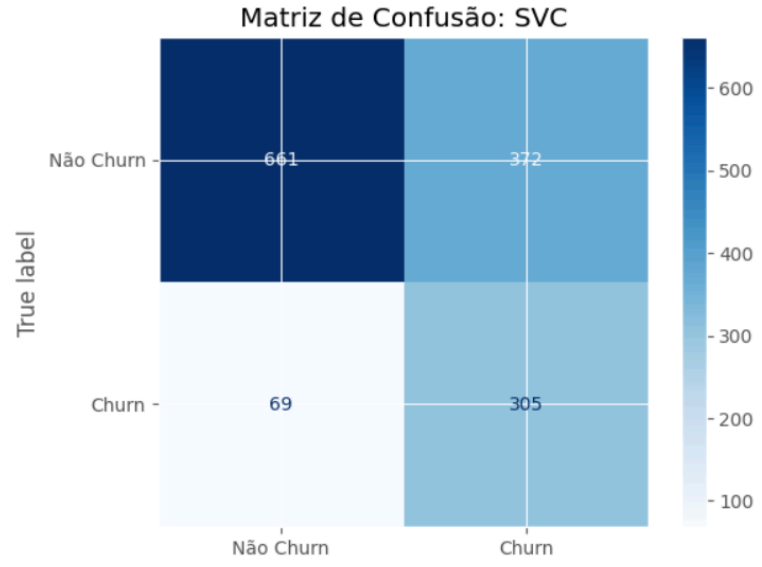
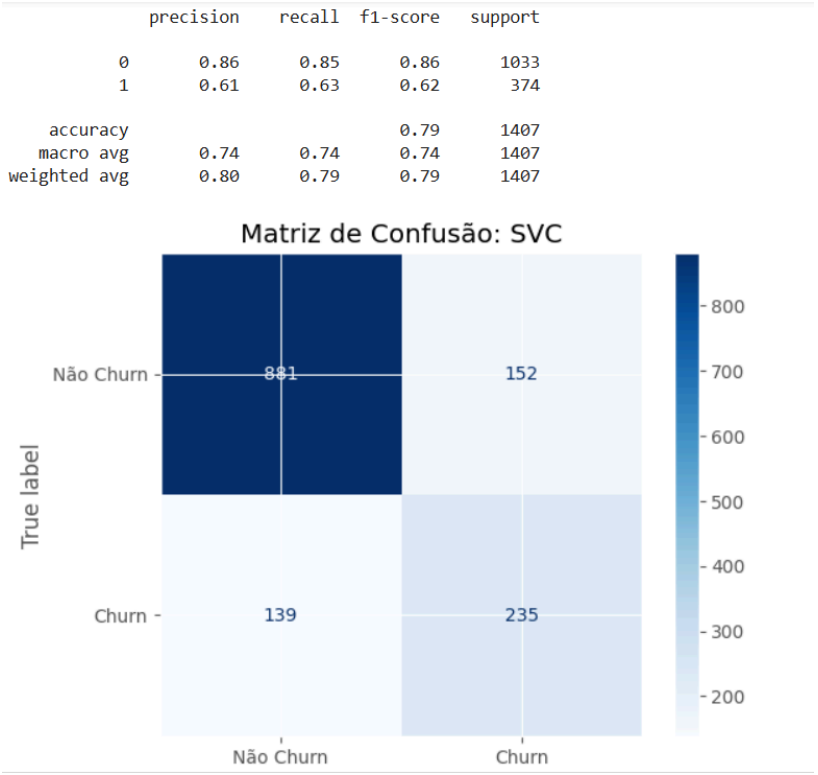
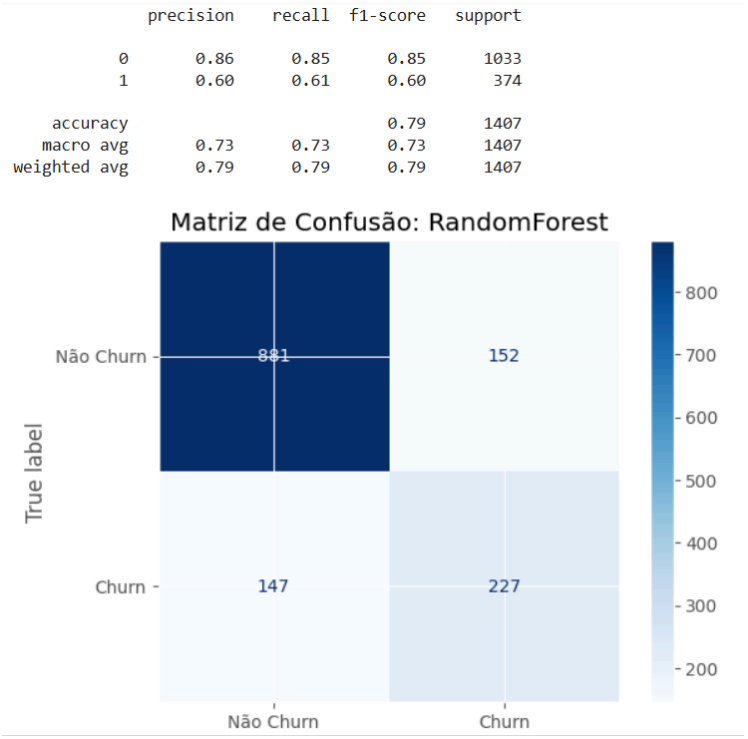


Imagem 6: Matriz de Confusão e Métricas de Avaliação do Método “SVC” com Undersampling Aleatório.



	precision	recall	f1-score	support
0	0.86	0.85	0.86	1033
1	0.61	0.63	0.62	374
accuracy			0.79	1407
macro avg	0.74	0.74	0.74	1407
weighted avg	0.80	0.79	0.79	1407

Imagem 7: Matriz de Confusão e Métricas de Avaliação do Método “SVC” com Undersampling (Tomek Links).



	precision	recall	f1-score	support
0	0.86	0.85	0.85	1033
1	0.60	0.61	0.60	374
accuracy			0.79	1407
macro avg	0.73	0.73	0.73	1407
weighted avg	0.79	0.79	0.79	1407

Imagem 8: Matriz de Confusão e Métricas de Avaliação do Método “Random Forest” com Undersampling.

Note que a diferença não foi significativa nas estatísticas, então é preferível usar o aprendizado sem as modificações das técnicas, a fim de preservar as características do banco de dados.

4 - ANÁLISE E CONCLUSÕES

Apesar de os modelos de Regressão Logística e SVC apresentarem a mesma acurácia global (0.80), a Regressão Logística demonstrou desempenho levemente superior nas métricas mais relevantes para a classe *Churn*, especialmente no Recall e no F1-score. Esses indicadores são cruciais em problemas de evasão de clientes, pois ajudam a identificar corretamente aqueles que estão prestes a cancelar o serviço. Essa vantagem torna a Regressão Logística mais adequada para apoiar ações proativas, como pesquisas de satisfação, com o objetivo de reduzir a evasão.

O modelo Random Forest também apresentou boa performance geral, com acurácia de 80%. No entanto, seu Recall para a classe Churn foi de apenas 0.52, o que significa que quase metade dos clientes que realmente iriam sair não foram identificados. Isso é preocupante, já que falsos negativos nesse contexto representam perda de oportunidades de retenção. O alto número de falsos negativos (181) reduz a eficácia do modelo em estratégias de retenção.

Embora todos os modelos avaliados apresentam acurácia semelhante (80%), essa métrica, sozinha, pode ser enganosa. Em problemas de churn, métricas específicas para a classe minoritária (Churn), como Recall e F1-score, são mais relevantes.

Dessa forma, a Regressão Logística se destaca como o modelo mais indicado, por aliar bom desempenho na detecção de churn, simplicidade, eficiência e fácil interpretação, facilitando a aplicação prática em estratégias de retenção.