CASE AQUISIÇÃO – PREDIÇÃO DE CLIENTES DE UM JORNAL DIGITAL

Filippo Petroli

Porto Alegre
2019

I. Definição

Visão geral do projeto

Pode-se dizer que otimizar processos é um objetivo de qualquer empresa. Seja a empresa de qual ramo for, ter seu desempenho maximizado com a menor quantidade de recursos pode ser um tiro certeiro no desenvolvimento do negócio.

Em setembro de 2017, com objetivo de acelerar a estratégia digital do Grupo RBS, foi criado o GaúchaZH, um jornal digital. Em 2018 foi registrado um crescimento de mais de 40% da carteira digital e recordes de audiência. A estratégia de aceleração no Grupo RBS é data driven e está em constante crescimento da maturidade analítica, ampliando a sua capacidade de predição e prescrição. Eles buscam cada vez mais entender o comportamento dos usuários para entregar melhores produtos e ampliar a carteira de assinantes. Em 2019, o objetivo é ampliar ainda mais a capacidade de aquisição de novos assinantes através de um modelo de predição.

Descrição do problema

O problema a ser resolvido para ampliar a capacidade de aquisição de novos clientes é entender o comportamento dos usuários da plataforma. Desta forma, pode-se então utilizar um algoritmo de classificação de machine learning para prever e rotular os usuários como possíveis assinantes ou não. Com isto, é possível saber quais clientes fazem valer o investimento de uma ligação ou um e-mail mais direcionado para a conversão, por exemplo.

Para tal, seguirá-se as seguintes etapas na construção do modelo:

- Avaliação e limpeza de dados
- Breve análise exploratória
- Pré-processamento
- Modelagem
- Tuning

Métricas

Para avaliar o desempenho do modelo será usado o f1-score. O f1-score é uma média entre precisão e recall. Foi escolhida esta métrica pois de acordo

com uma análise prévia, a base de dados é desproporcional e o f1-score é uma ótima métrica para estes casos.

II. Análise

Exploração dos dados

Para a solução deste problema, o Grupo RBS forneceu duas bases de dados com um conjunto de pessoas que assinaram e não assinaram a plataforma em um determinado mês.

As bases fornecidas são a Planilha_1.csv com dados de uso da plataforma e a Planilha_2.csv com dados demográficos.

Referente a Planilha_1.csv, as informações de uso são dos 60 dias para trás do mês para prospects (pessoas não assinantes) e 60 dias para trás relativos à data da compra para assinantes. A base conta com 5600 observações. As suas variáveis são:

- ID – identificação única de indivíduo;
- QT_TOTAL_HIT_PAYWALL – quantidade de Hit Paywall (hit paywall é quando a pessoa tenta ler uma matéria e não é assinante, recebe um pop-up para assinar);
- DIASNAVEGADOS – quantidade de dias navegados não importando a origem (Aplicativo ou Site);
- VISITAS_CAPA – quantidade de visitas à capa de GaúchaZH;
- NOTICIASLIDAS – quantidade de notícias lidas total;
- USOU_APP – se usou ou não usou algum aplicativo de GaúchaZH.

A Planilha_2.csv, com os dados demográficos, também conta com 5600 observações. As suas variáveis são:

- ID – identificação única de indivíduo;
- Perfil - Define se o indivíduo é ou não assinante.
- Gênero - F: Feminino, M: Masculino, I: Indefinido;
- PES_NASCIMENTO_DATA - Data de nascimento “dia.mes.ano”. Obs.: considera-se que a pessoa precisa ter mais de 18 anos para se tornar assinante e uma margem de até 100 anos para considerar que a pessoa ainda está viva;

- ATR_PF_GEO_RENDA_FAM - Renda em Salários mínimos por medida Geográfica, ‘SM’ significa salários mínimos.

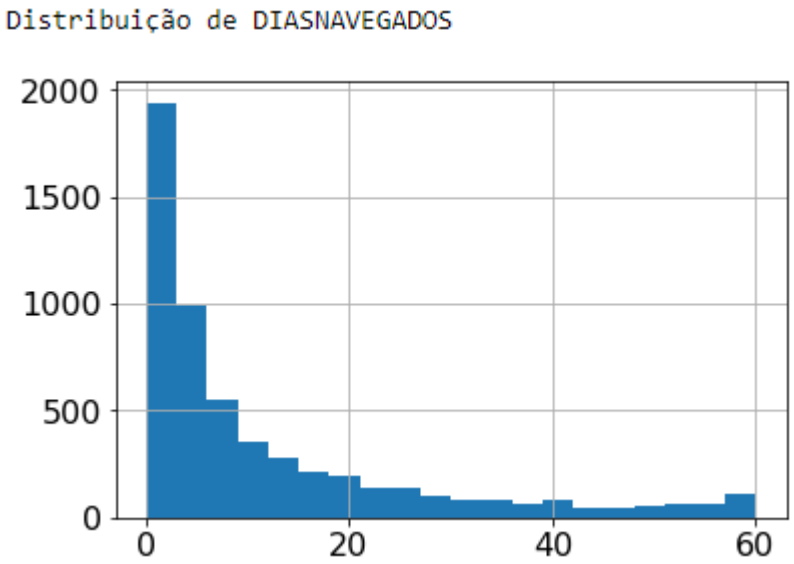
Após uma exploração inicial nestas bases de dados, foi constatado alguns problemas que podem dificultar o desenvolvimento do modelo. São eles:

- Os dados estão em dois datasets diferentes.
- O tipo de dado da coluna PES_NASCIMENTO_DATA está incorreto.
- PES_NASCIMENTO_DATA com datas inválidas.
- PES_NASCIMENTO_DATA e ATR_PF_GEO_RENDA_FAM com dados ausentes.

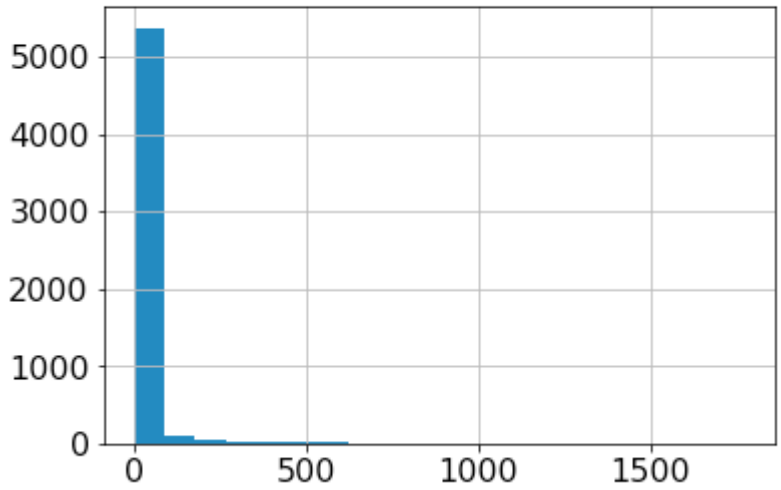
Cada problema foi tratado de forma individual. Primeiro tratou-se de unir as bases de dados através do ID correspondente de cada indivíduo. Logo após, modificou-se o tipo de dado da coluna PES_NASCIMENTO_DATA para o formato de data. Em seguida, foram desconsideradas datas de nascimento onde o usuário poderia ter menos de 18 anos ou maior de 100 anos. Nestes casos, as datas foram substituídas por ‘nan’. Para finalizar o tratamento dos problemas constatados, os valores ausentes foram preenchidos por ‘None’.

Visualização exploratória

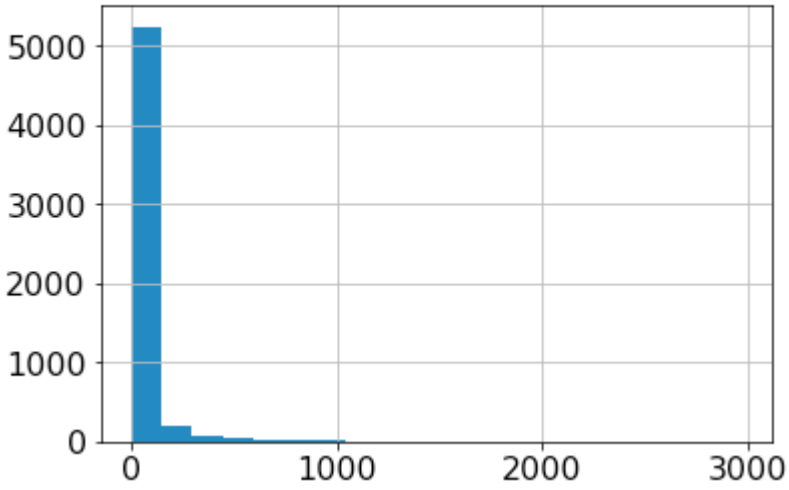
Agora com as bases unidas e algumas correções no dataset, foram vistas as distribuições de cada variável.

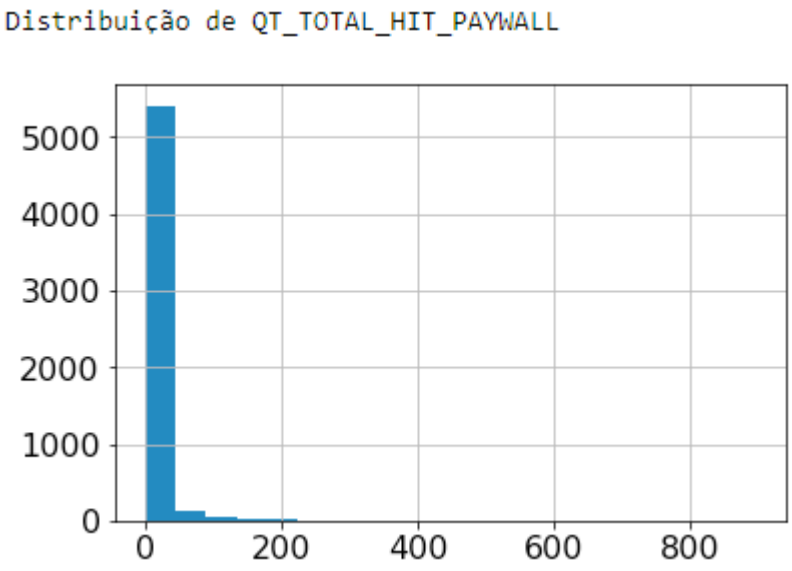


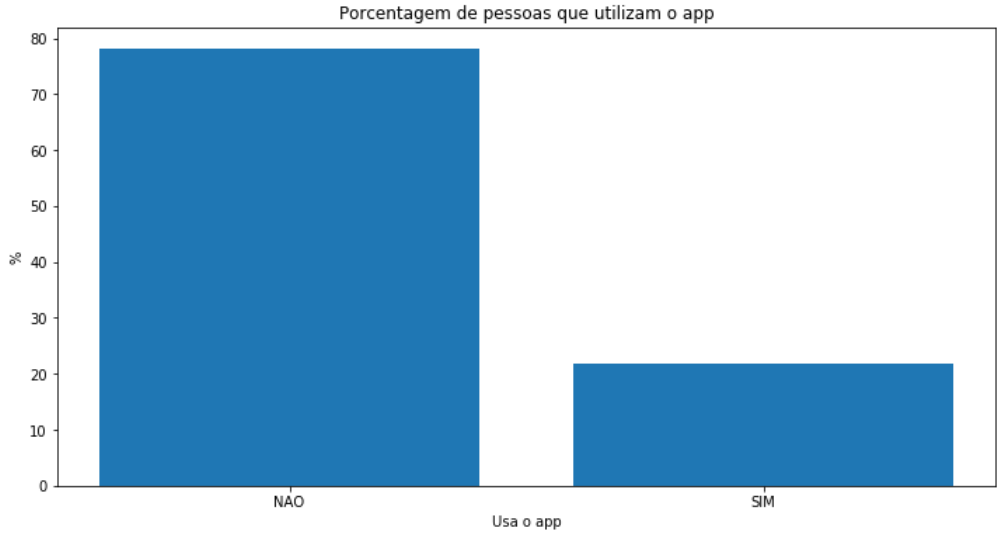
Distribuição de NOTICIASLIDAS

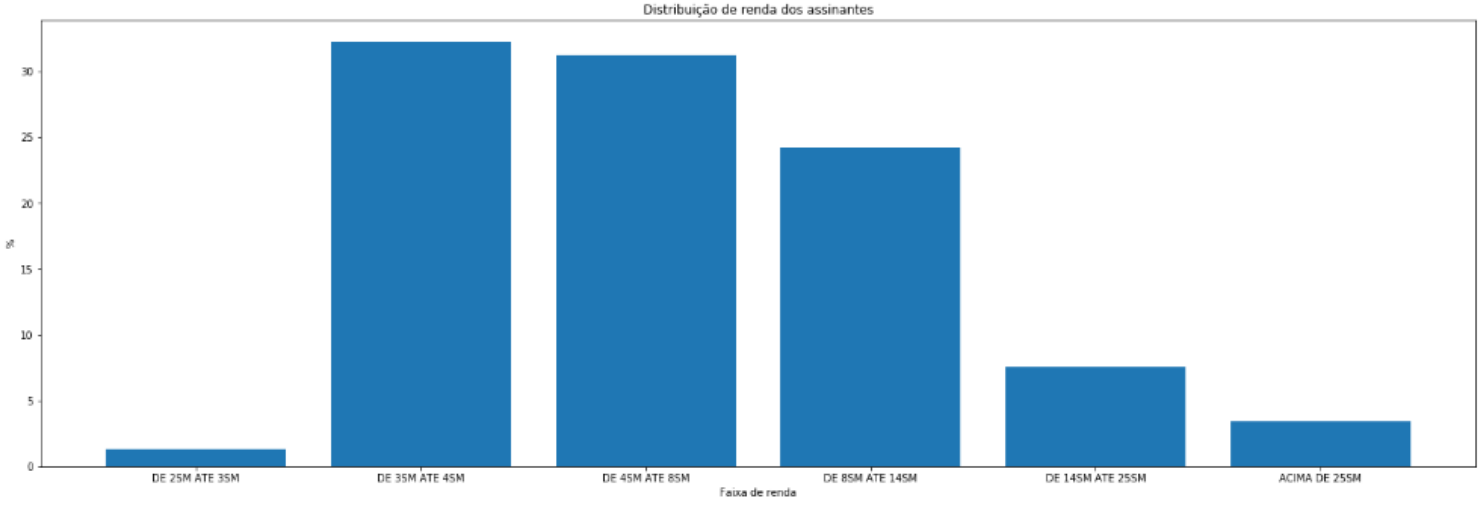


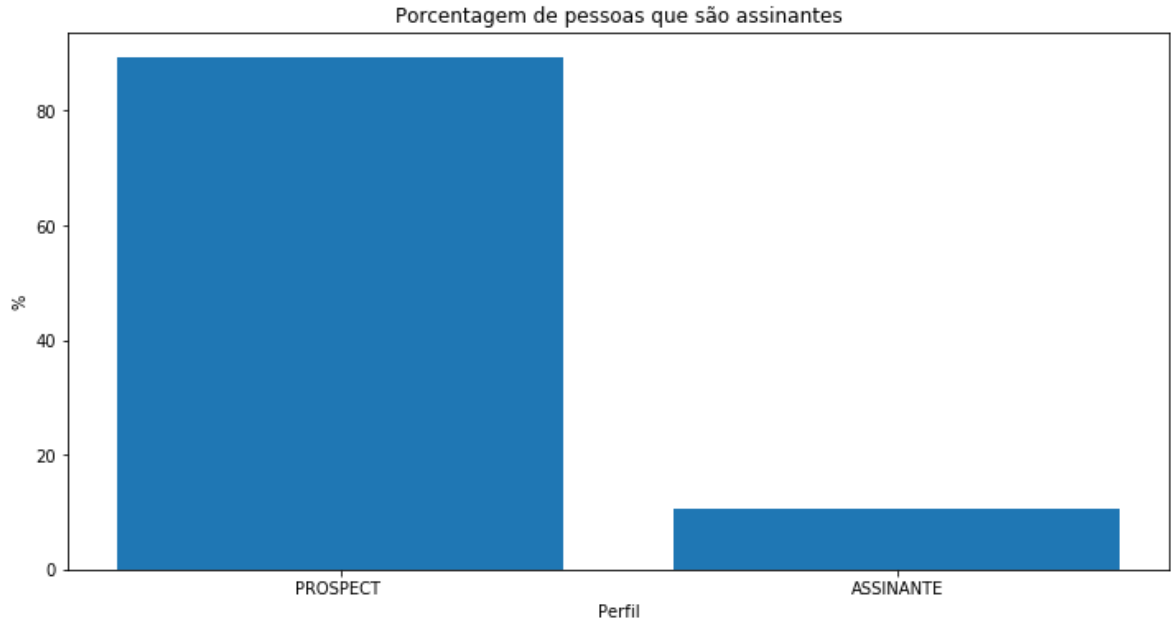
Distribuição de VISITAS_CAPA

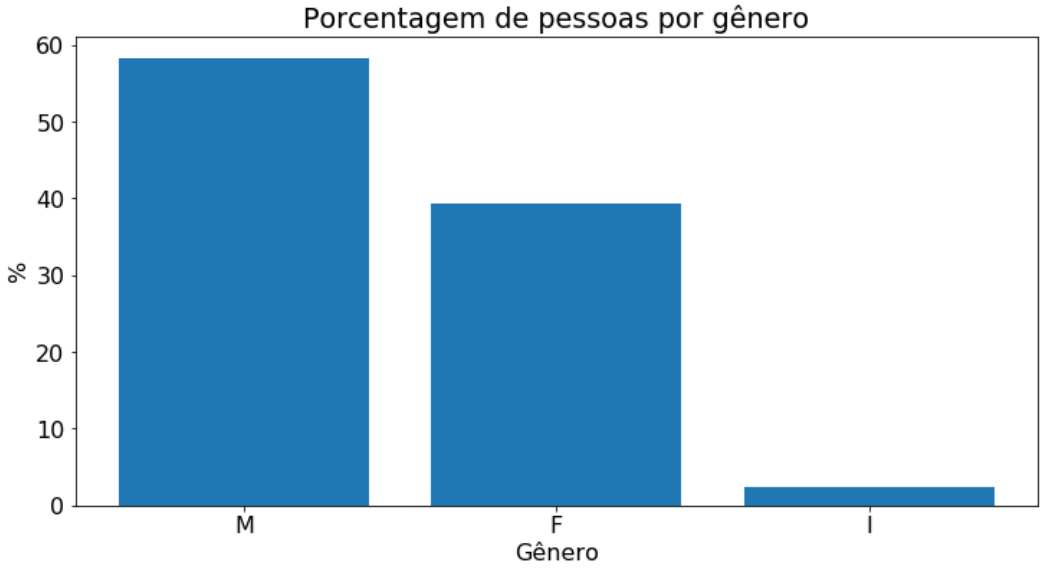



Distribuição de QT_TOTAL_HIT_PAYWALL
5000
4000
3000
2000
1000
0
0
200
400
600
800


Porcentagem de pessoas que utilizam o app
%
NAO
SIM
Usa o app


Distribuição de renda dos assinantes
%
DE 2SM ATE 3SM
DE 3SM ATE 4SM
DE 4SM ATE 8SM
DE 8SM ATE 14SM
DE 14SM ATE 25SM
ACIMA DE 25SM
Faixa de renda





Com a visualização das distribuições, foi possível observar que a maioria das variáveis não segue uma distribuição normal. Deste modo, tratou-se de criar uma função para trabalhar com outliers nas distribuições. Assim, os valores fora dos percentis entre 2,5 e 97,5 foram considerados outliers e substituídos pelas medianas correspondentes. Também ficou evidente o uso necessário da métrica f1-score, pois a base não está balanceada entre assinantes e não assinantes.

Ainda, visualizou-se as correlações das variáveis.

	QT_TOTAL_HIT_PAYWALL	DIASNAVEGADOS	NOTICIASLIDAS	\
QT_TOTAL_HIT_PAYWALL	1.000000	0.459818	0.061129	
DIASNAVEGADOS	0.459818	1.000000	0.539521	
NOTICIASLIDAS	0.061129	0.539521	1.000000	
VISITAS_CAPA	0.317927	0.666299	0.343202	

	VISITAS_CAPA
QT_TOTAL_HIT_PAYWALL	0.317927
DIASNAVEGADOS	0.666299
NOTICIASLIDAS	0.343202
VISITAS_CAPA	1.000000

Como observado na imagem anterior, não há fortes correlações entre as variáveis.

Algoritmos e técnicas

Para escolher em definitivo o modelo que será usado na resolução do problema, optou-se por testar uma série de algoritmos de machine learning com o intuito de selecionar o que fornecesse melhor resultado.

Os algoritmos escolhidos foram: Random Forest Classifier, Gradient Boosting Classifier e KNeighbors Classifier. Todos eles são plausíveis para apresentar soluções ao problema pois se adequam a categoria dos dados.

O Random Forest Classifier é um algoritmo o qual cria série de árvores de decisão e cria uma combinação delas para obter uma predição melhor e mais estável.

O modelo de Gradient Boosting é um algoritmo que utiliza vários pequenos outros modelos mais fracos para criar um único mais forte. Para isto, ele identifica as deficiências dos modelos fracos por gradientes.

Já o KNeighbors Classifier é um modelo que olha para um ponto, olha para seus pontos vizinhos mais próximos e determina a classe majoritária destes, deste modo, aplica sobre o ponto inicialmente visto.

Todos estes modelos são algoritmos que podem ser usados em casos de classificação, que é o objetivo do principal deste projeto.

Benchmark

Como benchmark, escolheu-se testar os dados em um algoritmo de árvore de decisão para se ter uma referência inicial do desempenho que pudesse obter como solução. Este teste foi realizado após o pré-processamento

dos dados, mencionado no tópico a seguir, devido a natureza dos dados não permitir aplicar o modelo de benchmark antes. Neste teste inicial, o f1-score ficou em 0.60.

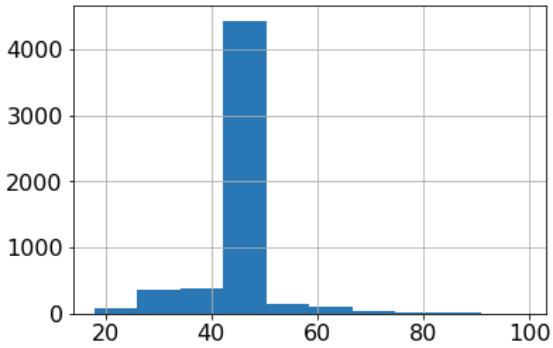
III. Metodologia

Pré-processamento de dados

Antes de aplicar o benchmark e iniciar a construção do modelo, algumas transformações são necessárias.

Iniciou-se esta etapa dividindo os dados entre features e target, passo primordial para construção de qualquer modelo.

Logo após, como é de conhecimento que algoritmos de machine learning não conseguem lidar bem com datas, decidiu-se transformar os dados da variável da data de nascimento em idade. Desta forma, terá-se valores inteiros e facilitará o uso e desempenho do modelo. Para isto, criou-se uma função que calcula a idade, fazendo a diferença entre o dia atual menos a data de nascimento e retornando uma idade em número inteiro. Para as datas com valores ausentes, o valor idade assumiu a média das idades. Nesta etapa, analisou-se também a distribuição da variável idade, ficando conforme a imagem a seguir.



Em seguida, como haviam muitos dados numéricos em diferentes proporções e grandezas, optou-se por normalizar estas variáveis. A normalização tem como objetivo transformar os dados numéricos de diferentes escalas em uma única escala, sem distorcer com as diferenças das variações dos números. Para isto, utilizou-se o método de MinMaxScaler, o qual transfora os dados em uma escada de 0 a 1.

Ainda, para o tratamento de variáveis categóricas, aplicou-se a técnica de one-hot encoding. Esta técnica transforma variáveis categóricas em variáveis binárias. Desta forma, algoritmos de machine learning conseguem fazer um trabalho melhor ao lidar com estes tipos de variáveis.

Por fim, foi realizado o split dos dados em dois datasets, um de treino e outro de teste, sendo o dataset de treino composto por 80% dos dados do dataset original e o de teste pelos outros 20%.

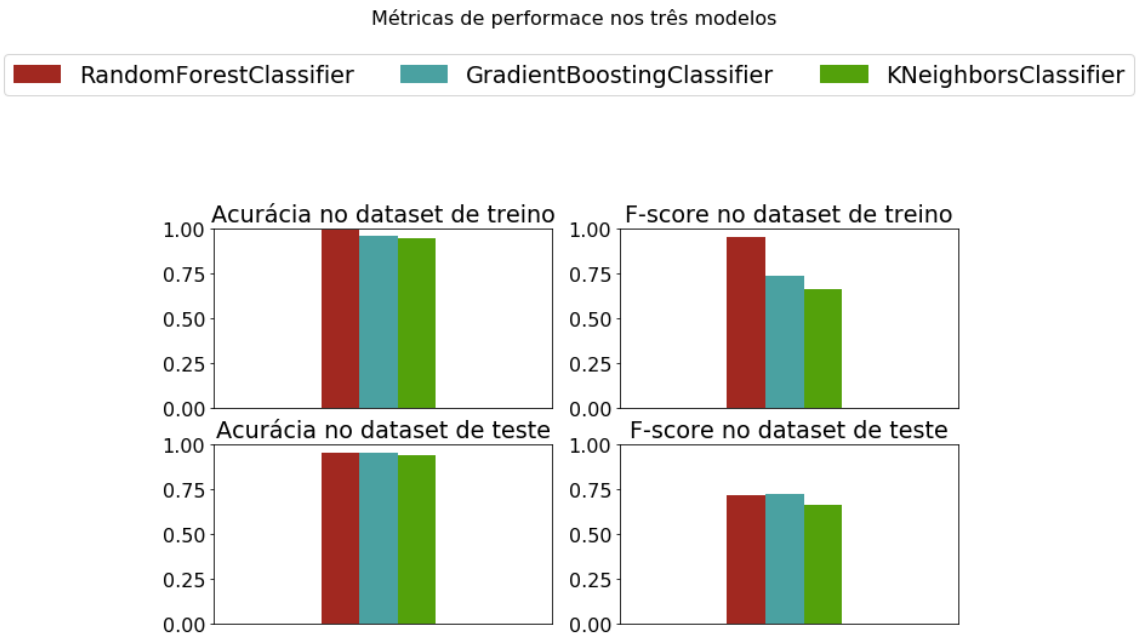
Implementação

Nesta etapa, com o intuito de automatizar o processo de escolha e treinamento do modelo, criou-se um pipeline. O pipeline consegue executar, de forma rápida e fácil, os três algoritmos e medir seu desempenho no estado inicial, sem qualquer alteração.

Na imagem a seguir, é possível observar os resultados obtidos com cada um dos três algoritmos:

```
{'RandomForestClassifier': {'acc_train': 0.9897321428571428,
  'acc_test': 0.9464285714285714,
  'f_train': 0.9491150442477876,
  'f_test': 0.7115384615384616},
 'GradientBoostingClassifier': {'acc_train': 0.9558035714285714,
  'acc_test': 0.9508928571428571,
  'f_train': 0.7367021276595745,
  'f_test': 0.7236180904522613},
 'KNeighborsClassifier': {'acc_train': 0.9455357142857143,
  'acc_test': 0.9375,
  'f_train': 0.6620498614958448,
  'f_test': 0.6568627450980392}}
```

Ainda, gerou-se um gráfico para facilitar a visualização.



Com estes resultados, foi possível definir o algoritmo escolhido para ser trabalhado na solução do problema. Como a métrica observada é a f1-score no dataset de teste, o algoritmo Gradient Boosting Classifier foi elencado para ser aprimorado.

Refinamento

Após escolher trabalhar com o Gradient Boosting Classifier, decidiu-se aperfeiçoar o algoritmo para tentar encontrar resultados mais satisfatórios. Para isto, a técnica utilizada foi de Grid Search. Esta técnica permite trocar e testar de forma fácil os parâmetros do algoritmo e retornar o que obtém melhor resultado de acordo com a métrica definida. Neste caso, foram trocados os valores dos parâmetros n_estimators , learning_rate e subsample do algoritmo. Com o uso do Grid Search foi possível elevar o f1-score para 0.73.

IV. Resultados

Modelo de avaliação e validação

Por fim, com o modelo já construído e otimizado, observou-se a métrica proposta inicialmente. O resultado obtido com o Gradient Boosting Classifier foi satisfatório, atingindo 0.73 em uma base de testes não utilizada para treinamento. Acredita-se que a solução encontrada está alinhada com o proposto inicialmente de classificar possíveis assinantes.

Como modelo final então, o Gradient Boosting Classifier foi construído com os parâmetros de acordo com a imagem a seguir.

```
 Modelo final
------
GradientBoostingClassifier(criterion='friedman_mse', init=None,
              learning_rate=0.05, loss='deviance', max_depth=3,
              max_features=None, max_leaf_nodes=None,
              min_impurity_split=1e-07, min_samples_leaf=1,
              min_samples_split=2, min_weight_fraction_leaf=0.0,
              n_estimators=500, presort='auto', random_state=None,
              subsample=0.5, verbose=0, warm_start=False)
```

Justificativa

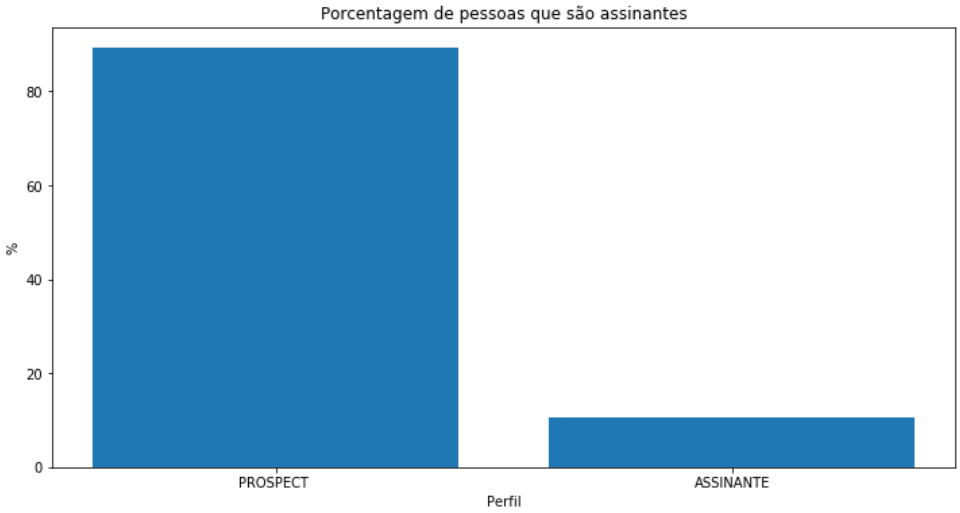
Em comparação aos modelos de benchmark, acredita-se que a escolha do Gradient Boosting Classifier foi adequada. Além do modelo ter apresentado melhores resultados que o benchmark (0.73 contra 0.60 da árvore de decisão), o modelo conseguiu bater os outros dois algoritmos testados, Random Forest Classifier e KNeighborsClassifier.

V. Conclusão

Forma livre de visualização

Um dos aspectos relevantes do projeto foi o uso da métrica de f1-score. Como foi visto anteriormente na análise exploratória e pode ser visto novamente na imagem a seguir, a base é desbalanceada nos dois rótulos da classificação.

Há muito mais não assinantes do que assinantes e quando isto ocorre, a métrica mais apropriada é o f1-score pois ela consegue lidar bem com esse problema de proporção.



Reflexão

Espera-se ao final do projeto que ele seja de grande valia para a predição de novos clientes do GaúchaZH. Desta forma, o time de aquisição poderá trabalhar de modo mais assertivo em suas decisões de negócio na captação de clientes. Acredita-se que ele já pode ser usado em produção, contudo, quando surgir novos dados, o modelo precisará passar por novos treinamentos.

É interessante observar, mais do que a forma de uso, a construção do modelo como um todo, passando por todas suas etapas desde a exploração inicial, pré-processamento de dados, benchmark e por fim o seu aperfeiçoamento. Em cada etapa foi possível ver toda a narrativa da construção até chegar ao resultado final, com um alto f1-score, mostrando-se um método maduro para ser empregado em, quem sabe, larga escala no negócio digital.

Melhorias

Ao final do projeto é possível ainda avaliar e mencionar alguns outros aspectos que podem contribuir com futuramente. Vale destacar, sempre, que quantos mais dados, melhor, obviamente. Não que o número de registros não

foi suficiente para modelar um bom algoritmo, mas quanto mais informações temos a disposição, mais bem treinado o modelo será.

Ainda, vale ressaltar que no aperfeiçoamento do modelo, no momento em que foi utilizado o Grid Search, apenas três parâmetros foram testados. Nesta fase, poderiam ser testados outros parâmetros presentes no Gradient Boosting Classifier que poderiam levar a um f1-score maior.

É importante mencionar que este modelo precisará de atualizações no futuro, conforme for testado em produção e avaliado no mundo real o seu acerto. Novos dados surgirão e será necessário treinar novamente o algoritmo para continuar performando bem.