

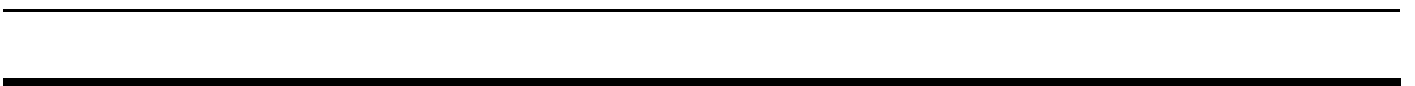
Projeto final

Nanodegree Engenheiro de Machine Learning

Rodrigo Domingos

12/03/2018

Melhorando a retenção de clientes na indústria de seguros.



Sumário

Nanodegree Engenheiro de Machine Learning	
Definição	1
Análise	4
Metodologia	10
Resultados	15
Conclusão	18
Referências	21

“O ramo de seguros Patrimoniais atingiu a marca de R\$ 9 bilhões, representando um crescimento de 6% comparado ao mesmo período no ano anterior”

Definição

Mercado Segurador Brasileiro

O Mercado de seguros no Brasil movimentou de janeiro de 2017 até novembro de 2017 R\$ 356 bilhões considerando todos os ramos com uma participação de aproximadamente 7% do PIB brasileiro. O ramo de seguros Patrimoniais atingiu a marca de R\$ 9 bilhões, representando um crescimento de 6% comparado ao mesmo período no ano anterior. *1

Ciclo de vida de uma apólice de seguro Patrimonial

O mercado de seguros brasileiro opera seguindo um fluxo comum, conforme demonstrado no fluxograma abaixo:

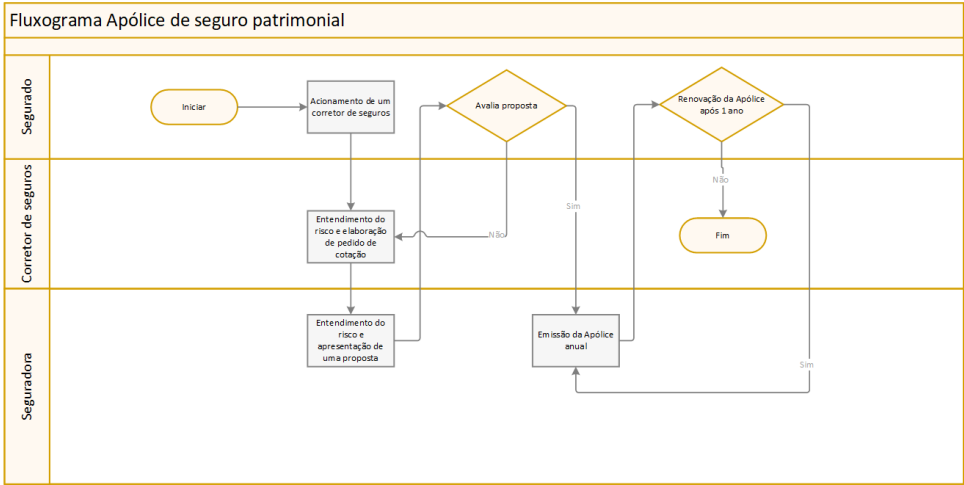


Figura 1 - Fluxo de cotação e emissão de apólices patrimoniais

Os 4 passos básicos do fluxo são:

- Cotação: Onde o Cliente junto ao corretor elaboram o perfil de coberturas necessárias.
- Proposta: Onde a seguradora envia uma proposta de valor de prêmio para assumir o risco.
- Emissão: Onde a seguradora emite a apólice de seguros, usualmente com vigência anual formalizando a aceitação do risco.
- **Renovação:** Ao final da vigência, a seguradora apresenta uma nova proposta a fim de dar continuidade por mais um ano ao contrato.

Desafio da retenção de clientes

A Renovação da apólice e consequente retenção do cliente, geralmente é baseada no valor da proposta apresentada e na experiência do cliente e corretor com a seguradora. A conquista de um novo cliente custo em média de 4 a 5 vezes mais do que manter a fidelidade de um cliente, pois isenta a empresa de reexecutar a fase de convencimento do cliente. Em 2017 aproximadamente 70 seguradoras disputaram uma fatia de mercado no ramo de seguros patrimoniais, numa disputa essencialmente baseada em preço e benefícios, embora seja verdade, nota-se que o relacionamento entre seguradora x corretor x cliente tem um peso significativo no momento da decisão da contratação ou renovação de um seguro.

Visão geral do projeto

O projeto “Melhorando a retenção de clientes na indústria de seguros” tem como objetivo analisar os dados históricos da carteira de clientes de uma seguradora a fim de encontrar padrões de comportamentos nos clientes que não renovaram suas apólices, com isto criar um modelo preditivo que aplicado as apólices que estão vigentes hoje, retornem a probabilidade de determinado cliente não renovar sua apólice ao final do contrato. Como o foco é identificar previamente o possível comportamento dos clientes este caso será endereçado por um modelo de aprendizagem de máquina de classificação binária baseado nas classes “NaoRenovou” para as contas que não renovaram e a “Renova” para os casos que renovaram. O modelo será alimentado com uma massa de dados estruturados, aplicará a classificação binária e o resultado será um arquivo no formato CSV com uma dentre as duas possíveis classes.

Isto possibilitaria a seguradora manter uma régua de comunicação e interação diferenciada com os clientes com alta probabilidade de não renovar, com esse cliente sentindo-se “Único” e o estreitamento no relacionamento com o cliente, espera-se uma melhoria na retenção no dos clientes e no **Índice de renovação** das apólices.

Métricas

O produto obtido através do modelo é uma lista das apólices que vigentes com a probabilidade de a conta ser renovada, esta probabilidade será transformada através de um limitador em duas possíveis respostas: “Renova” / “Não Renova”.

A métrica a ser utilizada para treinamento do modelo será “Log Loss” que penaliza as predições incorretas em ambas as classes (Renova(1)/Não Renova(0)).

$$-\frac{1}{N} \sum_{i=1}^N [y_i \log p_i + (1 - y_i) \log (1 - p_i)].$$

As métricas a serem consideradas para avaliação da qualidade das predições do modelo baseada na necessidade e impacto para o negócio (Retenção na Indústria de seguros) serão:

$$\text{True negative rate (Specificity): } TNR = \frac{TN}{TN+FP}$$

$$ACC: ACC = \frac{tp+tn}{tp+tn+fp+fn}$$

Assumindo como positiva a classe “Renova (1)”, desta forma torna-se mais importante a assertividade dos casos que não irão renovar; isto justifica-se pela natureza do problema em questão, onde as futuras ações a serem tomadas para os casos previstos como “Não Renova (0)” não terão impacto negativo caso eventualmente sejam aplicadas aos casos que deveriam ter sido classificados pelo modelo como “Renova (1)” ao contrário irá fortalecer a fidelidade do cliente.

Finalmente, a métrica usada para medir o impacto pós implantação do modelo e das ações tomadas com base no modelo é o Índice de renovação:

$$\text{Índice de renovação} = \frac{\text{Renovadas}}{\text{Apólices a vencer}}$$

“9 meses como massa para treinamento do modelo e inicialmente a previsão será realizada com dois meses de antecedência.”

Análise

Exploração dos dados

Para este projeto as informações selecionadas inicialmente foram pensando na experiência do cliente/corretor com a seguradora ao longo da vigência da apólice. Para isto foi utilizada técnicas de ETL (*Extraction Transformation and Load*) para processar estas informações desde os sistemas de origem até a massa de dados final.

Sistema de origem	Campo extraído	Transformação	Campo transformado	Conceito
Sistema de Emissão	IdConta	Não se aplica	IdConta	Identificador único da conta (Apólice_
Sistema de Emissão	AnoMes	Não se aplica	AnoMes	Ano e Mês da data final da vigência da apólice
Sistema de Emissão	Produto	Não se aplica	Produto	Nome do produto comercializado
SERASA	SaudeFinancCli	Soma pendencias financeiras agrupada por Cliente	SaudeFinancCli	Dados oriundos de consulta realizada no SERASA através do CNPJ do cliente
Sistema de Sinistro	ExpSinistroCI	Se não teve sinistro 0 Se teve sinistro indenizado 1 Se teve sinistro não indenizado 2	ExpSinistroCli	Sumarização da experiência de sinistro do cliente
Sistema de Sinistro	ExpSinistroCorretor	Se não teve sinistro 0 Se teve sinistro indenizado 1 Se teve sinistro não indenizado 2	ExpSinistroCorr	Sumarização da experiência de sinistro do corretor
Sistema de Emissão	Qtd Emissões	(Qtd Emissões / Qtd Cotações) referente aos últimos 6 meses	IndFechCorr	Cálculo utilizado na indústria de seguros para medir a conversão das propostas enviadas.
Sistema de Emissão	Qtd Cotações			
Sistema de assistência	ExpAss24	Não acionou 0 Acionou 1	ExpAss24	Monitoramento de acionamento do benefício de assistência 24h pelo cliente
Sistema de call center	ExpCallCenterC	Soma ligações no call center	ExpCallCenterCli	Monitoramento da quantidade de interações entre o cliente e a central de atendimento.
Sistema de Emissão	Uf	Não se aplica	Uf	Estado do Cliente
Sistema de Emissão	PremioBrutoK	Não se aplica	PremioBrutoK	Valor pago pelo cliente para contratar a apólice
Calculado	Resultado	Não se aplica	Resultado	Verificação se a conta foi renovada ou não

Estão sendo utilizados 9 meses como massa para treinamento do modelo e inicialmente a previsão será realizada com dois meses de antecedência. Há um cuidado para que os dados de treino não sobreponham os dados de teste, ex.: As apólices emitidas em 201701 serão as mesmas que irão vencer um ano depois ou seja 201801, ao incluir os dados de 201701 ao tentar prever os resultados de 201801 haverá um “Vazamento de informação” que afetará a qualidade do modelo.

Divisão lógica dos dados

	Treinamento									Intervalo para ações		Previsão
2011701	201702	201703	201704	201705	201706	201707	201708	201709	201710	201711	201712	201801

Amostra dos dados:

IdConta	AnoMes	Produto	SaudeFinancCli	ExpSinistroCli	ExpSinistroCorr	IndFechCorr	ExpAss24	ExpCallCenterCli	UF	PremioBrutoK	Resultado
100	201701	COMPREENSIVO EMPRESARIAL	4	2	2	0	0	1	SAO PAULO	15	NaoRenovou
200	201701	COMPREENSIVO EMPRESARIAL	1	2	2	0	1	1	RIO DE JANEIRO	12	NaoRenovou
300	201701	COMPREENSIVO EMPRESARIAL	4	1	2	0	0	1	RIO DE JANEIRO	18	NaoRenovou
400	201701	COMPREENSIVO EMPRESARIAL	5	2	2	0,333333333	1	1	MINAS GERAIS	10	NaoRenovou
600	201701	COMPREENSIVO EMPRESARIAL	3	2	2	0	1	2	PARANA	9	NaoRenovou
700	201701	COMPREENSIVO EMPRESARIAL	4	2	2	0,333333333	1	1	RIO GRANDE DO SUL	9	NaoRenovou
1000	201701	COMPREENSIVO EMPRESARIAL	5	1	2	0,5	1	1	RIO DE JANEIRO	12	NaoRenovou
1100	201701	COMPREENSIVO EMPRESARIAL	5	2	2	0	1	2	PARANA	6	NaoRenovou
1200	201701	COMPREENSIVO EMPRESARIAL	5	2	2	1	0	3	PARANA	11	NaoRenovou

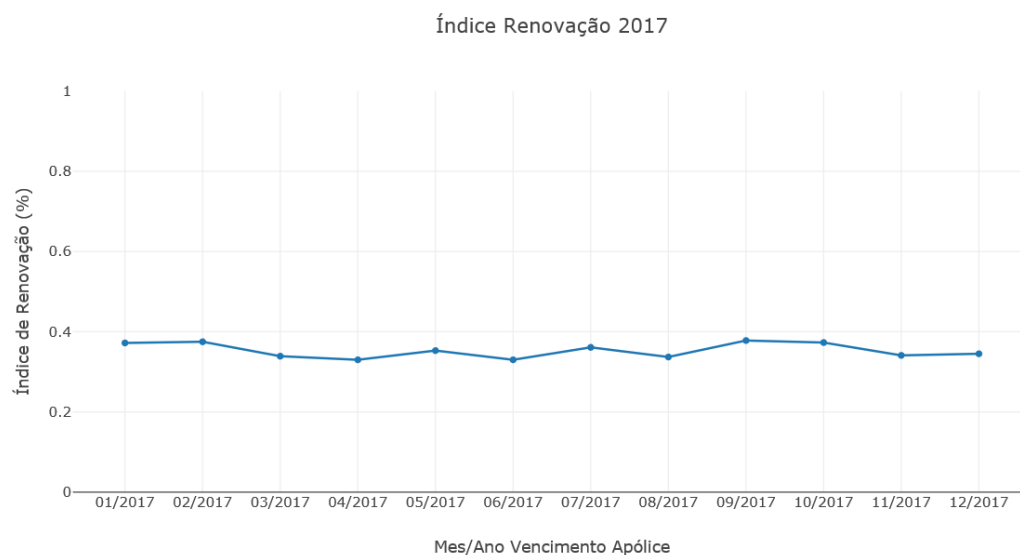
Estatística dos dados:

	IdConta	AnoMes	SaudeFinancCli	ExpSinistroCli	ExpSinistroCorr	ExpAss24	ExpCallCenterCli	PremioBrutoK
count	1.300000e+04	13000.000000	13000.000000	13000.000000	13000.000000	13000.000000	13000.000000	13000.000000
mean	6.500500e+05	201713.769231	2.751769	1.249385	1.372385	0.496077	1.736231	10.527769
std	3.752921e+05	25.399807	1.587511	0.724092	0.755536	0.500004	1.012444	5.785003
min	1.000000e+02	201701.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000
25%	3.250750e+05	201704.000000	1.000000	1.000000	1.000000	0.000000	1.000000	6.000000
50%	6.500500e+05	201707.000000	3.000000	1.000000	2.000000	0.000000	2.000000	11.000000
75%	9.750250e+05	201710.000000	4.000000	2.000000	2.000000	1.000000	3.000000	16.000000
max	1.300000e+06	201801.000000	5.000000	2.000000	2.000000	1.000000	3.000000	20.000000

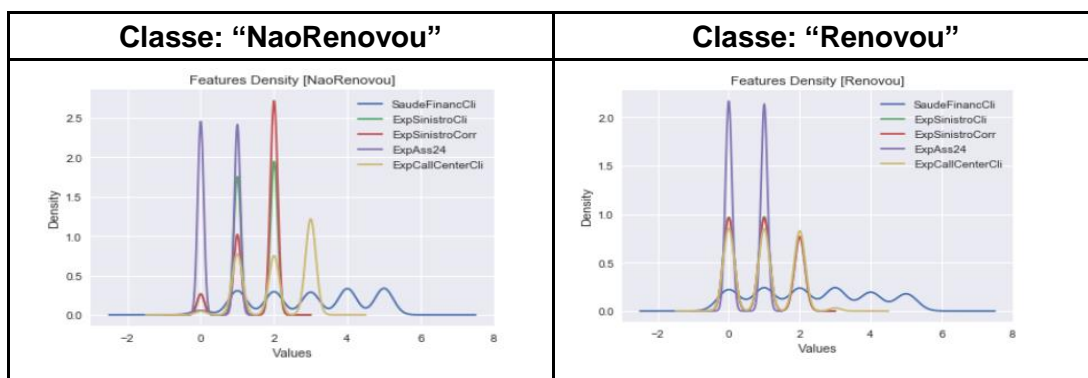
Devido ao pré-processamento realizado sobre os dados usando ETL os efeitos de possíveis “Outliers” e dados nulos estão devidamente tratados. Baseado na média pode-se observar que não existem grandes variações comparando os valores máximos e mínimos.

Visualização exploratória

Os resultados históricos demonstram que o índice de renovação mensal é em média 36% conforme gráfico abaixo:



Numa análise detalhada, é possível observar um comportamento diferente das variáveis que compõem a massa de dados para os casos que historicamente renovaram ou não as apólices:



Esta visualização nos dá indícios visuais de que há um comportamento padrão ou pelo menos diferenciado para os casos “NaoRenovou”, por exemplo as duas informações referentes à sinistros com valores (2 – Sinistros ocorridos mas que não foram indenizados) tem uma forte concentração na classe “NaoRenovou” assim como Experiência do cliente

com *call center* (3 – Quantidade de ligações) possui uma concentração bem maior na classe “NaoRenovou”.

Algoritmos e técnicas

Para resolução deste problema está sendo utilizado um modelo preditivo supervisionado de classificação binária, cujo objetivo será prever a maior probabilidade entre duas possíveis classes “NaoRenovou” assumindo o valor (0) e “Renovou” assumindo o valor (1).

Baseado em experiências anteriores e os bons resultados observados em competições de ciência de dados o algoritmo que será utilizado será o Extreme Gradient Boosting “XGBOOST” (Distribuição nativa [link](#)), “O XGBoost é uma biblioteca otimizada e distribuída projetada para ser altamente eficiente, flexível e portátil. Ele implementa algoritmos de aprendizado de máquinas sob o framework Gradient Boosting.” Além da excelente precisão nos resultados é extremamente versátil em termos de plataforma e performático em termos de tempo de treinamento e otimização dos recursos computacionais disponíveis. Um outro ponto positivo deste modelo é o auto controle sobre os dados Nulos e sobre massa de dados desbalanceadas, Para o processamento e manipulação de dados serão utilizadas as bibliotecas do framework Scikit-learn [link](#).

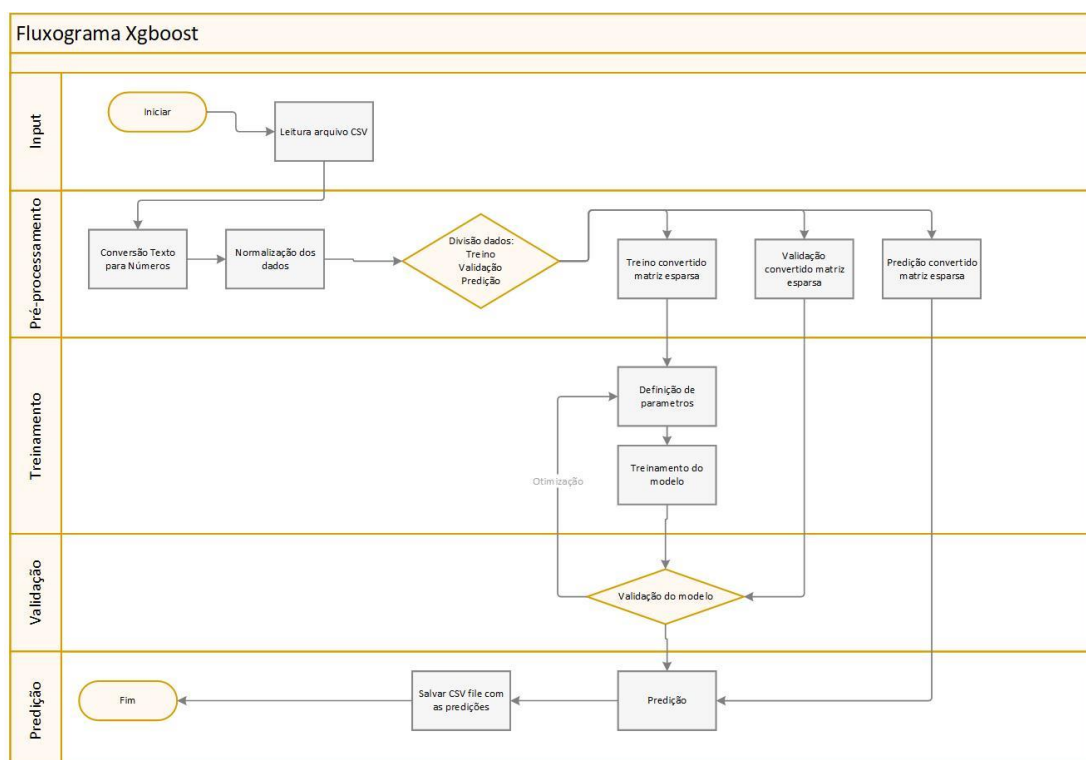


Figura 2 - Fluxograma Xgboost

Especificações técnicas:

- *Sistema operacional*
 - *Microsoft Windows 10*
- *Frameworks & Bibliotecas*
 - Anaconda
 - Python 3.5.4
 - Numpy 1.12.1
 - Pandas 0.22
 - Pandasql 0.7.3
 - Scikit-learn 0.19.1
 - Plotly 2.2.2
 - Qgrid 0.3.3
 - Seaborn 0.8.1
 - Matplot 2.2.12
 - Py-xgboost 0.6
 - Libxgboost 0.6

Benchmark

Para esta solução não há um valor de alguma solução de mercado para ser usada como comparativo a fim aferir a assertividade da solução aqui proposta, porém baseado na realidade de negócio, ao identificar com sucesso no mínimo 70% dos casos que potencialmente irão deixar a seguradora com dois meses de antecedência (*"True negative"*), seria suficiente para dar condições de atuar de forma diferenciada nestes casos e se revertidos resultariam em resultados financeiros positivos e no fortalecimento da marca na visão do cliente. Uma outra comparação que será realizada é baseada na performance de um algoritmo básico de árvore de decisão cujo o classificador sobre a mesma massa de dados apresentou os seguintes resultados:

Destaque para as os resultados obtidos de TNR e ACC.

```
#Benchmark using a basic Decision tree classifier
from sklearn import tree
clf = tree.DecisionTreeClassifier()
clf = clf.fit(x_train, y_train)
bench_pred = clf.predict(x_valid)
cm=confusion_matrix(y_valid, bench_pred)
print("Benchmark result:")
from pandas_ml import ConfusionMatrix
cm = ConfusionMatrix(list(y_valid['Resultado']), list(bench_pred))
print("Class statistics: ")
cm.print_stats()

Benchmark result:
Class statistics:
population: 1000
P: 341
N: 659
PositiveTest: 346
NegativeTest: 654
TP: 288
TN: 601
FP: 58
FN: 53
TPR: 0.844574780059
TNR: 0.911987860395
PPV: 0.832369942197
NPV: 0.918960244648
FPR: 0.0880121396055
FDR: 0.167630057803
FNR: 0.155425219941
ACC: 0.889
F1_score: 0.838427947598
MCC: 0.753941874425
informedness: 0.756562640453
markedness: 0.751330186845
prevalence: 0.341
LRP: 9.59611689756
LRN: 0.170424658804
DOR: 56.3070917372
FOR: 0.0810397553517
```

“Nesta fase é aplicado o modelo sobre uma massa de dados que não possui a variável resposta e o modelo realiza as previsões com base nos valores de entrada gerando duas colunas adicionais contendo a probabilidade de renovação.”

Metodologia

Pré-processamento de dados

Conforme demonstrado anteriormente, os dados são oriundos de um processo de ETL, assim os efeitos de dados em nulos e *outliers* já foram devidamente tratados.

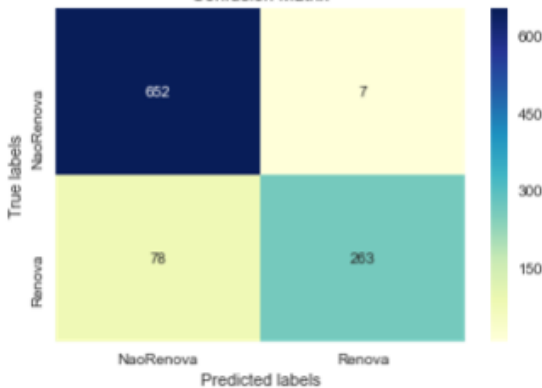
Neste projeto devido a utilização do algoritmo XGBOOST alguns procedimentos ainda serão necessários para adequação da massa de dados à entrada esperada pelo modelo:

- **Label Encoding:** Processo onde as colunas que possuem dados em forma de texto terão seus valores convertidos numa escala de [0,1]
- **Normalização:** Processo onde os dados numéricos serão convertidos numa escala de [0,1] a fim de padronizar a escala de todas as informações em uma escala comum.
- **Remoção de identificadores:** Processo onde as colunas com identificadores são isoladas da massa de dados (Para serem utilizadas posteriormente), a fim de criar um modelo generalista.
- **Divisão dos dados:** Processo onde a massa de dados é dividida entre: Treino, Validação e Predição.
- **Matrix Esparsa:** Alguns modelos esperam receber uma entrada de dados com a estrutura de matriz esparsa como é o caso do *xgboost*, neste os valores distintos são transpostos em colunas e quando houver a ocorrência um valor será preenchido e quando não houver será adicionado o número 0 como valor padrão.

Treinamento, validação e refinamento

Nesta fase acontece a definição de parâmetros que iniciam com os valores padrões seguidos de rodadas de treinamento e validações iterativas na busca pela por melhores resultados; abaixo 3 amostras do processo de refinamento:

- *Modelagem inicial*

<pre>logger.info("Start data training") # Parameters xgb_params = {'eta': 0.05 , 'gamma': 0 , 'min_child_weight':1 , 'max_delta_step':0 , 'subsample':1 , 'colsample_bytree ':0.8 , 'colsample_bylevel':1 , 'lambda': 1 , 'alpha':1 , 'scale_pos_weight':1 , 'max_depth': 4 , 'objective': 'binary:logistic' , 'eval_metric': 'logloss' , 'seed': 99 , 'silent': True} # Model training watchlist = [(d_train, 'train'), (d_valid, 'valid')] model = xgb.train(xgb_params , d_train, 1000 , watchlist , maximize=False , verbose_eval=50 , early_stopping_rounds=10) logger.info("Finish data training")</pre>	<pre>[0] train-logloss:0.662719 valid-logloss:0.662713 Multiple eval metrics have been passed: 'valid-logloss' will Will train until valid-logloss hasn't improved in 10 rounds. [50] train-logloss:0.257725 valid-logloss:0.250675 [100] train-logloss:0.199975 valid-logloss:0.187635 [150] train-logloss:0.184219 valid-logloss:0.173843 [200] train-logloss:0.177886 valid-logloss:0.169531 [250] train-logloss:0.173559 valid-logloss:0.167645 Stopping. Best iteration: [271] train-logloss:0.172242 valid-logloss:0.167084</pre>									
<pre>Class statistics: population: 1000 P: 341 N: 659 PositiveTest: 270 NegativeTest: 730 TP: 263 TN: 652 FP: 7 FN: 78 TPR: 0.771260997067 TNR: 0.98937784522 PPV: 0.974074074074 NPV: 0.893150684932 FPR: 0.01062215478 FDR: 0.0259259259259 FNR: 0.228739002933 ACC: 0.915 F1_score: 0.860883797054 MCC: 0.812185223144 informedness: 0.760638842287 markedness: 0.867224759006 prevalence: 0.341 LRP: 72.6087138668 LRN: 0.231194789774 DOR: 314.058608059 FOR: 0.106849315068</pre>	<p>Confusion Matrix</p>  <table><tr><th></th><th>NaoRenova</th><th>Renova</th></tr><tr><th>True labels NaoRenova</th><td>652</td><td>7</td></tr><tr><th>True labels Renova</th><td>78</td><td>263</td></tr></table>		NaoRenova	Renova	True labels NaoRenova	652	7	True labels Renova	78	263
	NaoRenova	Renova								
True labels NaoRenova	652	7								
True labels Renova	78	263								

Resumo:

LogLoss: 01670

ACC: 0.915 (Assertividade geral do modelo considerando a predição das duas classes)

TNR (Specificity): 0.9893

Os resultados obtidos usando os valores padrões foram usados como base para a procura de melhores resultados.

- Modelagem intermediária

<pre>logger.info("Start data training") # Parameters xgb_params = {'eta': 0.2 , 'gamma': 0 , 'min_child_weight':1.2 , 'max_delta_step':0 , 'subsample':1 , 'colsample_bytree ':0.8 , 'colsample_bylevel':1 , 'lambda': 1 , 'alpha':1 , 'scale_pos_weight':1 , 'max_depth': 6 , 'objective': 'binary:logistic' , 'eval_metric': 'logloss' , 'seed': 99 , 'silent': True} # Model training watchlist = [(d_train, 'train'), (d_valid, 'valid')] model = xgb.train(xgb_params ,d_train, 1000 ,watchlist ,maximize=False ,verbose_eval=50 ,early_stopping_rounds=10) logger.info("Finish data training")</pre>	<pre>[0] train-logloss:0.570556 valid-logloss:0.570926 Multiple eval metrics have been passed: 'valid-logloss' will Will train until valid-logloss hasn't improved in 10 rounds. [50] train-logloss:0.164755 valid-logloss:0.165266 Stopping. Best iteration: [62] train-logloss:0.160758 valid-logloss:0.16465</pre>									
<p>Class statistics: population: 1000 P: 341 N: 659 PositiveTest: 288 NegativeTest: 712 TP: 273 TN: 644 FP: 15 FN: 68 TPR: 0.800586510264 TNR: 0.977238239757 PPV: 0.947916666667 NPV: 0.904494382022 FPR: 0.0227617602428 FDR: 0.0520833333333 FNR: 0.199413489736 ACC: 0.917 F1_score: 0.868044515103 MCC: 0.814264337216 informedness: 0.777824750021 markedness: 0.852411048689 prevalence: 0.341 LRP: 35.1724340176 LRN: 0.204058213876 DOR: 172.364705882 FOR: 0.0955056179775</p>	<p>Confusion Matrix</p>  <table><tr><th></th><th>Predicted labels NaoRenova</th><th>Renova</th></tr><tr><th>True labels NaoRenova</th><td>644</td><td>15</td></tr><tr><th>Renova</th><td>68</td><td>273</td></tr></table>		Predicted labels NaoRenova	Renova	True labels NaoRenova	644	15	Renova	68	273
	Predicted labels NaoRenova	Renova								
True labels NaoRenova	644	15								
Renova	68	273								

Resumo:

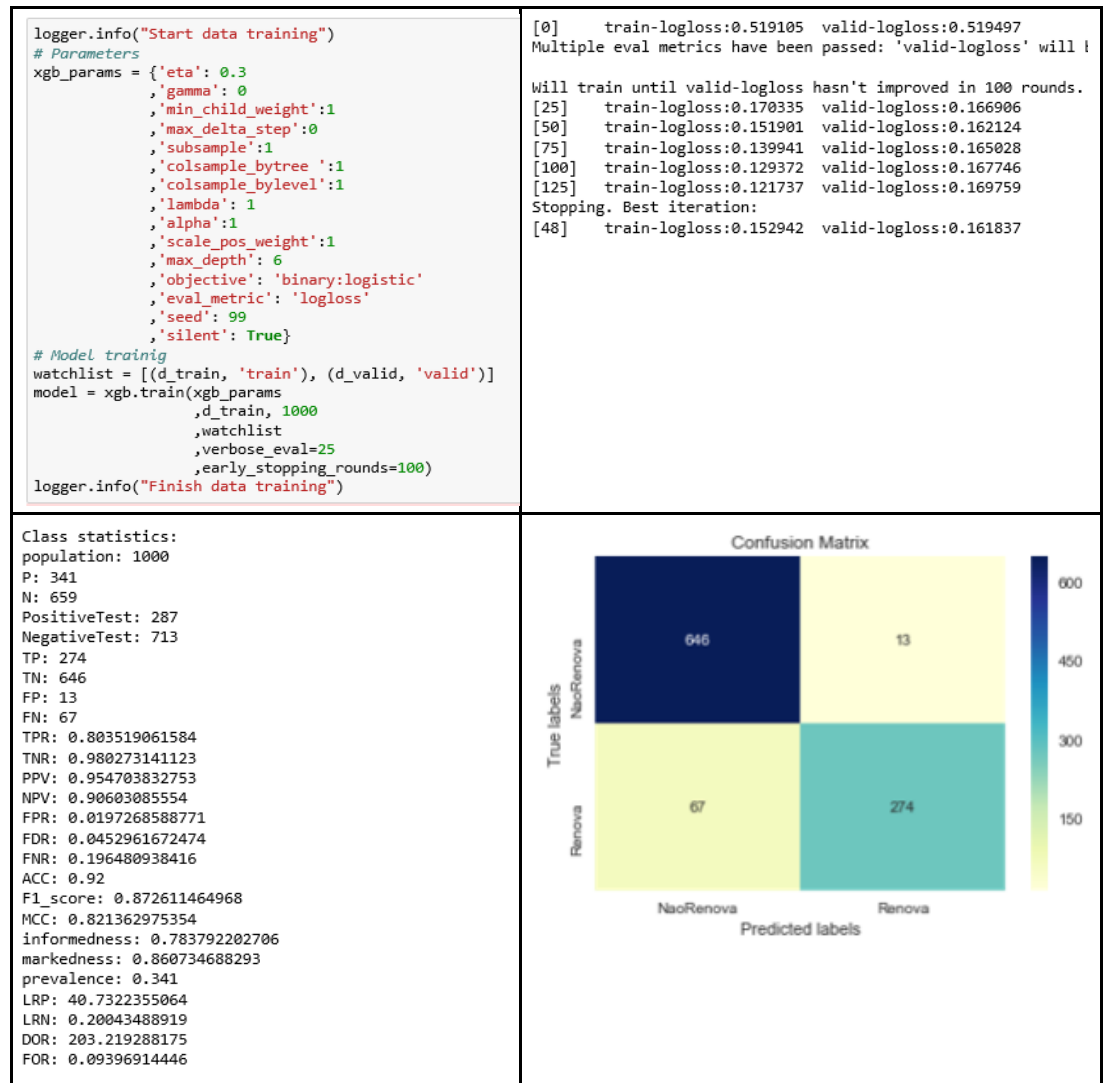
LogLoss: 01646

ACC: 0.917 (Assertividade geral do modelo considerando a predição das duas classes)

TNR (*Specificity*): 0.9772

O processo usado para otimização do modelo foi a escolha manual dos novos valores dos parâmetros, num modelo intermediário o TNR (Assertividade dos casos que não renovaram) diminui, porém a métrica ACC aumentou tornando o modelo mais genérico e robusto.

- Modelagem Final



Resumo:

LogLoss: 01618

ACC: 0.912 (Assertividade geral do modelo considerando a predição das duas classes)

TNR (*Specificity*): 0.9802

Após diversas tentativas esta é a combinação de parâmetros que apresentaram os resultados mais balanceados de ACC e TNR.

Abaixo o range de parâmetros utilizados no processo de refinamento:

	Mín	Melhor	Máx
eta	0.05	0.3	0.7
max_depth	4	6	7
colsample_bytree	0.8	1	1

Predição

Nesta fase é aplicado o modelo sobre uma massa de dados que não possui a variável resposta e o modelo realiza as previsões com base nos valores de entrada gerando duas colunas adicionais contendo a probabilidade de renovação e a predição aplicando a regra de limite $0 < 0.5 < 1$

Ex:

Produto	SaudeFinancCli	ExpSinistroCli	ExpSinistroCorr	IndFechCorr	ExpAss24	ExpCallCenterCli	Uf	PremioBrutoK	Resultado	prediction%	prediction
ENSIVO ESARIAL	2	2	2	0,333333333	0	1	RIO GRANDE DO SUL	3	NaN	0.060356	0.0
ENSIVO ESARIAL	2	1	1	0	1	1	SAO PAULO	9	NaN	0.363099	0.0
ENSIVO ESARIAL	3	1	2	0,5	0	3	RIO GRANDE DO SUL	20	NaN	0.003467	0.0
ENSIVO ESARIAL	1	2	2	0	0	1	SAO PAULO	17	NaN	0.299381	0.0
ENSIVO ESARIAL	5	2	2	0,5	1	2	MINAS GERAIS	3	NaN	0.076292	0.0

“A confiabilidade das previsões do modelo supera 85%, o que é maior do que o valor definido como “benchmark (50%)”. Com esta taxa de assertividade é possível atuar sobre estas contas de forma diferenciada com a segurança de não estar colocando esforços onde eventualmente não haveria necessidade.”

Resultados

Modelo de avaliação e validação

Com base no desempenho apresentado o modelo pode ser considerado validado para o objetivo proposto.

Abaixo o demonstrativo de algumas simulações utilizando massas de dados de treino, validação e predição diferentes com o mesmo conjunto de parâmetros definidos na modelagem final

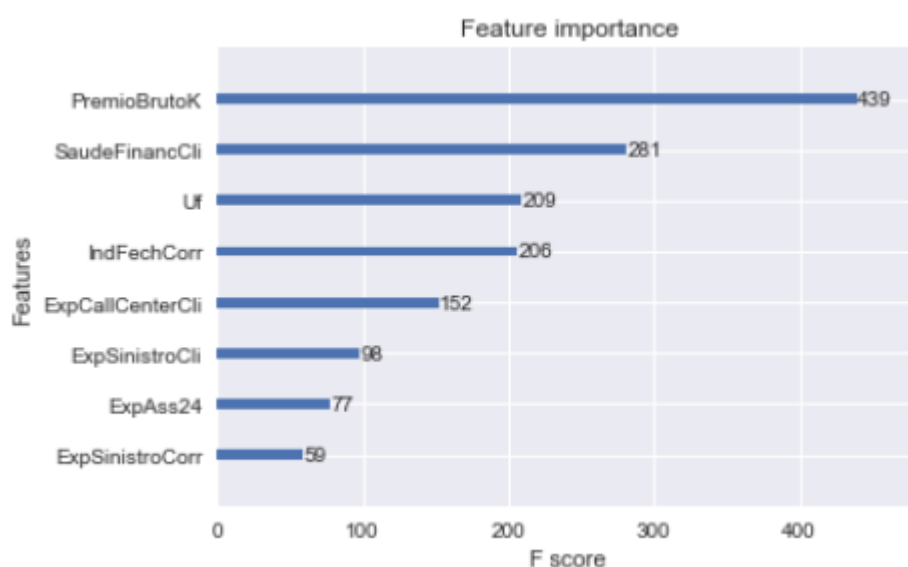
Configurações	Validação
Running the model with the following configurations: 2018-01-23 15:33:15.527614 Start training: 02/2017 End training : 10/2017 Validation : 11/2017 Prediction : 01/2018	
Running the model with the following configurations: 2018-01-23 16:02:49.944478 Start training: 01/2017 End training : 09/2017 Validation : 10/2017 Prediction : 12/2017	
Running the model with the following configurations: 2018-01-23 16:07:32.332763 Start training: 02/2017 End training : 11/2017 Validation : 12/2017 Prediction : 01/2018	

Justificativa

A confiabilidade das previsões do modelo supera 85%, o que é maior do que o valor definido como benchmark de negócio (70%) e. Com esta taxa de assertividade é possível atuar sobre estas contas de forma diferenciada com a segurança de não estar colocando esforços onde eventualmente não haveria necessidade. Ainda podemos observar que a performance do modelo escolhido Xgboost foi superior ao modelo classificador usado como benchmarks:

Métrica	Decision Tree	Xgboost
TNR	0.91	0.98
ACC	0.88	0.91

Embora seja um modelo preditivo é possível extrair algumas informações descritivas que podem auxiliar na identificação relativa das variáveis mais importantes para o modelo. Abaixo uma extração realizada do modelo usando o recurso de “*Feature Importance*”:



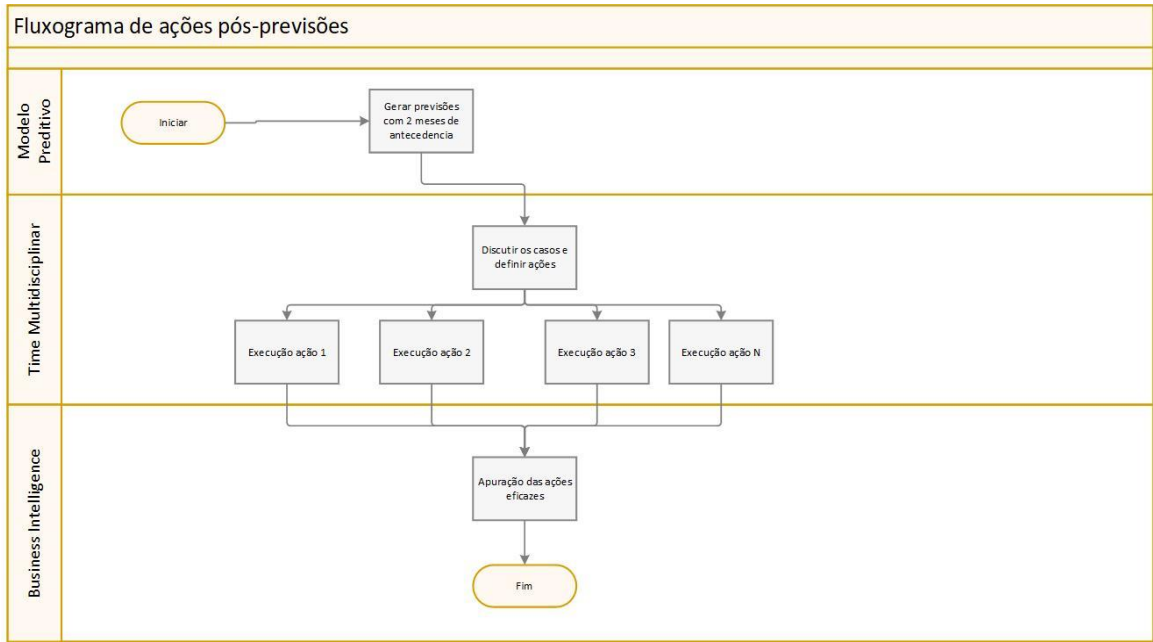
A visualização acima nos dá uma ideia da ordem de importância na construção da árvore de decisão do modelo e abre a possibilidade de uma análise descritiva sobre cada variável e os processos de negócios relacionados. É importante observar que visualizações como esta podem confirmar ou negar como foi o caso deste projeto a hipótese das variáveis mais importantes. Havia uma suspeita inicial de que as variáveis relacionadas à sinistros seriam

as mais importantes, como podemos ver segundo o modelo elas foram classificadas em penúltimo e último lugar na escala de importância relativa.

Conclusão

Forma livre de visualização

A indicação preditiva das contas com baixa probabilidade de renovação é o ponto inicial do projeto “Melhorando a retenção de clientes na indústria de seguros”, a partir desta lista com as previsões é necessário a definição de um plano de ação por uma equipe multidisciplinar, a execução destas ações e a monitoramento das que causaram impacto positivo, conforme fluxo sugerido abaixo:



Idealmente a equipe multidisciplinar deve ser formada por:

- **Vendas**
- **Contas a pagar**
- **Marketing**
- **Subscrição**
- **Controle de risco**

O processo de definição das ações deve levar em consideração as informações relativas à cada conta dentro do escopo de variáveis da massa de dados, Ex.: Cliente possui alta concentração de pendencias financeiras ou Cliente possui diversas ligações ao Call center. Este tipo de análise possibilitará a definição de uma ação mais apropriada para cada caso.

Após a realização das ações, preferencialmente o departamento de *Business Intelligence* irá apurar a eficácia das ações para cada conta, lembrando que cada ação executada deve ser registrada para que o estudo possa ser alimentado.

Reflexão

A elaboração deste projeto foi muito construtiva, inicialmente havia o desafio de pensar em um problema real e meu foco estava na indústria de seguros, durante uma análise do mercado de seguros percebi o potencial deste mercado e refleti sobre a interação das seguradoras com seus segurados de forma geral e notei que aumentar o Índice de renovação das apólices anuais trariam um impacto financeiro significativo para as seguradoras e que existem uma lacuna entre o relacionamento entre seguradora e cliente que se preenchido pode influenciar na decisão do cliente.

Uma vez definido o problema, discuti com especialistas a fim de identificar variáveis que pudessem ter relação ou explicar o fenômeno dos clientes não renovarem suas apólices e esta discussão resultou na definição da massa de dados utilizada neste projeto.

A coleta, tratamento e manipulação das informações desde seus sistemas de origem até a geração de um arquivo CSV foi realizada usando técnicas de ETL, esta fase também endereçou as questões de dados nulos e *outliers*.

A escolha do modelo foi baseada em experiências anteriores e resultados observados em competições de ciência de dados; a modelagem inicial foi realizada com valores padrões e já resultou em uma boa performance, após diversas rodadas de treinamento com diferentes combinações de parâmetros e valores a modelagem final foi definida.

O modelo está configurado e pronto para receber os dados de entrada conforme especificado e gerar um arquivo CSV contendo a probabilidade de as contas serem renovadas.

O ponto mais interessante observado no projeto foi a comprovação de que a hipóteses das variáveis explicativas foram capazes de explicar o fenômeno da Não Renovação, e que estas variáveis foram definidas em conjunto com especialistas o que prova o sucesso da sinergia do conhecimento humano com recursos de aprendizagem de máquina; já as duas fases que consumiram mais tempo foram a de ETL e a otimização dos parâmetros, o que demandou um conhecimento bem específico, cuidado e paciência ao longo da execução.

De forma geral, o projeto atendeu as expectativas do objetivo pelo qual foi desenvolvido; é capaz de prever com grau satisfatório de assertividade das contas que não irão renovar e consegue se adaptar a novas entradas de dados, vale ressaltar que a indicação previa das contas é só o início do processo, as ações a serem desenvolvidas sobre estas contas são tão importantes quanto possuir previamente esta informação.

Por fim, este trabalho me possibilitou criar um projeto completo de *Machine Learning*, desenvolvendo minhas habilidades de comunicação com público não técnico,

aprimoramento dos conhecimentos obtidos ao longo do programa Nanodegree, e a confiança de solucionar problemas/oportunidades utilizando aprendizagem de máquina.

Melhorias

A rápida evolução em AI está gerando novas possibilidades a cada dia, e não me espantaria que em um curto espaço de tempo um outro algoritmo consiga resultados melhores do que os obtidos neste projeto; porém tomando como base as técnicas utilizadas acredito que existam alguns pontos que poderiam ser melhorados num futuro próximo:

- **Nova variável Ação tomada**

A ideia é após 12 meses a partir do modelo entrar em produção, retroalimentar a massa de dados com ações tomadas a fim de que elas possam melhorar as previsões

- **Substituir CSV**

A ideia é substituir os arquivos CSV's de entrada e saída por conexões diretas com uma base de dados garantindo mais segurança da informação que é extremamente confidencial estratégica.

Referências

1. Fonte: Udacity (ML) <https://classroom.udacity.com/nanodegrees/nd009/syllabus/core-curriculum>
2. Fonte: CNSEG - <http://www2.susep.gov.br/menuestatistica/SES/principal.aspx>
3. Fonte: dmlc/Xgboost <https://github.com/dmlc/xgboost/blob/master/doc/parameter.md>
4. Fonte: Wikipédia https://en.wikipedia.org/wiki/Sensitivity_and_specificity
5. Fonte: Wikipédia https://en.wikipedia.org/wiki/Loss_functions_for_classification
6. Fonte: Xgboost Doc. http://xgboost.readthedocs.io/en/latest/python/python_api.html
7. Fonte: Analytics Vidhay <https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/>
8. Fonte: ScikitLearn <http://scikit-learn.org/stable/>
9. Fonte: Plotly python <https://plot.ly/python/>
10. Fonte: Seaborn <https://seaborn.pydata.org/>
11. Fonte: Qgrid <https://github.com/quantopian/qgrid>