UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITÉCNICA DA USP
PROGRAMA DE EDUCAÇÃO CONTINUADA

ROBERTO SPADIM

Estimação de probabilidade de *default* com aplicação de modelos de aprendizado de máquina em empréstimos imobiliários

São Paulo
2018

ROBERTO SPADIM

Estimação de probabilidade de *default* com aplicação de modelos de aprendizado de máquina em empréstimos imobiliários

Versão original

Monografia apresentada à Escola Politécnica da Universidade de São Paulo pelo Programa de Educação Continuada, para conclusão do curso de Engenharia Financeira.

Orientador: Prof. Dr. André Cury Maiali

São Paulo

2018

Agradecimentos

Aos professores do PECE-Poli USP, em especial: Dr. André Cury Maiali, orientador e professor na matéria de derivativos, Dr. Bruno Augusto Angélico professor na matéria de séries temporais e Dr. Oswaldo Luiz do Valle Costa professor em várias matérias em especial na matéria de carteiras, que se dedicaram para transferir todo conhecimento e conteúdo durante o curso.

Ao amigo e professor Dr. Humberto Brandão por me incentivar a conhecer e explorar a área de aprendizado de máquina.

À minha namorada Ana Paula Pazzin Curiel por me incentivar o estudo de economia e finanças.

Resumo

SPADIM, Roberto. **Estimação de probabilidade de *default* com aplicação de modelos de aprendizado de máquina em empréstimos imobiliários**. 2018. 63 f. Escola Politécnica Universidade de São Paulo, São Paulo.

Este trabalho explora a utilização de modelos de aprendizado de máquina para estimar a probabilidade de *default*[1] em um conjunto de dados de empréstimos imobiliários para clientes "desbancarizados"[2].

A falta de um modelo referência para este problema leva ao uso de modelos de aprendizado de máquina. No desenvolvimento das etapas de modelagem é utilizada a validação cruzada como ferramenta de avaliação das variáveis e modelos. Os modelos de aprendizado de máquina utilizados foram do tipo *Gradient Boosted Trees*, que agregam diversas árvores de decisão utilizando adição sequencial de modelos com uso de *boosting*[3].

Os dados foram fornecidos pela empresa *Home Credit* utilizando a plataforma de competições em aprendizado de máquina *Kaggle*. A avaliação final das estimativas dos modelos foi executada pela plataforma.

Palavras Chaves: Aprendizado de Máquina, Classificação Binária, Probabilidade de Default, Risco de Crédito, Validação Cruzada, Empréstimos Imobiliários

[1] Ocorrência de evento de crédito

[2] Clientes que não possuem um histórico de crédito ou conta em instituições financeiras

[3] reamostragem e ponderação das amostras para redução do erro

Abstract

SPADIM, Roberto. **Estimação de probabilidade de *default* com aplicação de modelos de aprendizado de máquina em empréstimos imobiliários**. 2018. 63 f. Polytechnic School, University of São Paulo, São Paulo.

This paper explores the use of machine learning models to estimate the probability of default[4] in a data set of home loans to unbanked[5] customers.

The lack of a known reference model to solve this problem leads to the use of machine learning models. During the modeling steps, cross-validation is used as a tool to evaluate variables and models. The *Gradient Boosted Trees* models were used, which reduce the error adding sequential models with *boosting*[6].

Data were provided by *Home Credit* using the *Kaggle* machine learning platform. The final evaluation was performed by the platform.

Key Words: Machine Learning, Binary Classification, Default Probabilities, Credit Risk, Cross-Validation, Real Estate Loans

[4] Occurrence of credit event

[5] Those adults without an account at a bank or other financial institution

[6] Resampling and sample weighting

Lista de figuras

Lista de tabelas

Lista de abreviaturas e siglas

AI/IA	*Artificial Intelligence*, Inteligência Artificial
AUC	*Area Under the Curve*, área sob a curva
CART	*Classification And Regression Tree*, modelo de árvore de decisão para classificação e regressão
CV	*Cross-Validation*, validação cruzada
CSV	*Comma-Separated Values*, arquivo texto com separação dos pontos por vírgula, regulamentado pelo RFC 4180
GBDT/GBM/GBT	*Gradient Boosted Tree*
LGB	*LightGBM*
ML	*Machine Learning*, aprendizado de máquina
MSE	*Mean Squared Error*, erro quadrático médio
ROC	*Receiver Operating Characteristic*, característica de operação do receptor
ROC AUC	*Receiver Operating Characteristic, Area Under the Cruve*, área sobre a curva da característica de operação do receptor, métrica para avaliação de estimação de classificador binário
TP	Verdadeiro positivo, *True Positive*
TN	Verdadeiro negativo, *True Negative*
FP	Falso positivo, *False Positive*
FN	Falso negativo, *False Negative*
TPR	Taxa de verdadeiro positivo, *True Positive Rate*
TNR	Taxa de verdadeiro negativo, *True Negative Rate*
FPR	Taxa de falso positivo, *False Positive Rate*
FNR	Taxa de falso negativo, *False Negative Rate*

XGB *XGBoost*

Sumário

[7] De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

1 Introdução

1.1 Motivação

Muitas pessoas "desbancarizadas"[1] lutam para obter empréstimo. Infelizmente, credores não confiáveis se aproveitam dessa população oferecendo juros altíssimos ou calendários de pagamento financeiramente inviáveis. Credores confiáveis não conseguem oferecer propostas de crédito por falta de enquadramento de produto. Por fim, os clientes não são devidamente atendidos e as empresas perdem oportunidades de bons negócios.

1.2 Objetivo

A proposta deste trabalho é solucionar o problema de estimação da probabilidade de *default*[2] em operações de financiamento imobiliário de clientes "desbancarizados".

A solução proposta faz uso dos modelos de classificação binária, com ajuste aos dados por aprendizado supervisionado. Modelo de classificação binária é um estimador de classe, que utiliza um estimador de probabilidade condicional para uma única classe onde um parâmetro de nível de corte de probabilidade estimada é utilizado para estimar a classe. Aprendizado supervisionado é o processo de ajuste de dados de entrada e saída a um modelo.

Os modelos utilizados são do tipo *Gradient Boosted Trees* (GBT). Neste modelo, árvores de decisão são agregadas pelo método de *boosting*[3]. Para estimar a capacidade de generalização do modelo é utilizada a validação cruzada.

A métrica de performance utilizada será a *ROC AUC*[4], que mede a qualidade das estimativas de um estimador sem a necessidade da avaliação do nível de corte de probabilidade.

A estimativa da probabilidade de *default* é condicionada à variáveis que representam a situação macroeconômica, localização do bem financiado, qualidade do bem financiado, qualidade socio-econômico-financeira do cliente e informações do financiamento como: valor do bem, valor do financiamento e calendário de pagamento. Em problemas de gestão

[1] Clientes que não possuem um histórico de crédito ou conta em instituições financeiras
[2] Ocorrência de evento de crédito
[3] Reamostragem e ponderação das amostras para redução do erro
[4] Área sobre a curva da característica de operação do receptor

de risco de crédito, o cálculo da probabilidade de *default* é um dos fatores da **perda esperada**.

A perda esperada é utilizada para gerenciamento do risco de crédito, ela é definida pela soma da multiplicação de seus fatores: **probabilidade de *default*** (PD), **exposição no *default*** (EAD) e **perda dado o *default*** (LGD). Nos empréstimos bancários a perda esperada varia ao longo do tempo por diversas razões como: quitação de parcela, recuperação de perdas e atualização da probabilidade de *default*. Neste trabalho, foi estimada apenas a probabilidade de *default* para o instante da avaliação da proposta de financiamento. Os fatores LGD e EAD não serão estimados.

Dado o uso de modelos de aprendizado de máquina e otimização, não serão utilizadas técnicas que aumentam a complexidade de modelos e técnicas que "estressam os dados".

A importância prática deste trabalho é constatada pela ocorrência de uma competição remunerada, onde uma empresa de crédito imobiliário, *Home Credit*, disponibilizou os dados em parceria com a plataforma de competições de aprendizado de máquina *Kaggle*.

No trabalho os modelos foram aplicados aos dados desta competição e a métrica de avaliação da estimativa foi calculada pela plataforma *Kaggle*. Com objetivo de evitar procedimentos que podem influenciar na melhoria dos modelos com uso do placar da competição, foi limitado o envio de 7 (sete) estimativas. As 7 (sete) estimativas enviadas representam 2 (dois) modelos com 3 (três) estimativas cada, e uma estimativa a titulo de curiosidade representando a média das estimativas de todos os modelos enviados. Na "vida real" a espera pelo resultado do valor real da estimativa pode demorar meses ou anos até a quitação do financiamento.

1.3 Plataforma de competição Kaggle

Kaggle é uma plataforma *em núvem* para: hospedagem de base de dados, ambiente computacional para modelagem e sistema para competições de aprendizado de máquina.

No sistema de competição é fornecido um conjunto de serviços e regras: contextualização do problema e objetivo da competição; regras da competição como limites de envio de estimativas, número máximo de participantes por equipe, utilização de bases de dados externas, tipos de modelos e técnicas permitidas, tempo máximo de execução dos modelos, linguagem de programação permitida e outras regras específicas; conjunto de dados para

modelagem, análise e estimação; explicações gerais sobre os dados; explicação do sistema de ranqueamento dos competidores com uso de métricas de avaliação das estimativas; ambiente de simulação e modelagem; forum de discussão dos participantes; placar da competição e ambiente para envio e avaliação das estimativas.

Algumas competições são patrocinadas por empresas que buscam uma melhoria de processos internos, em geral elas remuneram os competidores e fornecem bases de dados e explicações sobre os problemas. Esta prática fomenta a participação de competidores experientes do meio acadêmico e da indústria. Devido ao ambiente proporcionado, a plataforma se tornou uma referência em compartilhamento de conhecimento sobre casos práticos do uso de aprendizado de máquina.

Durante uma competição a plataforma disponibiliza para todos participantes um placar chamado "público". O placar "púbico" mostra o resultado da melhor avaliação das equipes em uma amostra pequena do conjunto de dados. Ao final da competição cada equipe deve escolher 2 (duas) das estimativas enviadas para classificação em um placar final chamado placar "privado". O placar "privado" avalia as estimativas em todo conjunto de dados. Além do placar geral existe um placar privado e individual por equipe, neste pode-se avaliar cada conjunto de estimativas enviado com a métrica do placar "público" e ao final da competição a métrica do placar "privado".

1.4 Estrutura do trabalho

Os procedimentos são explicados em suas etapas nos capítulos deste trabalho, foi considerado que o leitor tem conhecimento básico de modelos de regressão e otimização.

Este trabalho está dividido em sete capítulos. No primeiro capítulo esta introdução. No segundo capítulo uma revisão bibliográfica dos temas relevantes para compreensão da métrica *ROC AUC*, validação cruzada e modelos de árvores de decisão utilizados. No terceiro capítulo uma abstração das etapas do processo de modelagem. No quarto capítulo a aplicação da teoria aos dados da competição. No quinto capítulo a avaliação dos resultados da modelagem e os resultados da competição. No sexto capítulo a conclusão do trabalho. Por fim, no sétimo capítulo ideias gerais para trabalhos futuros.

2 Revisão Bibliográfica

2.1 Conceitos básicos

Neste capítulo são apresentados os conceitos básicos para as etapas de modelagem. O texto das subseções 2.1.1, 2.1.2 e 2.1.3 foram baseados em (PRATI, 2008), (KOHAVI, 1995), (XGBOOST, 2016) e (MICROSOFT, 2018).

Em geral os modelos para estimação de probabilidade são desenvolvidos de duas formas. A primeira é a modelagem conhecida por *model-driven*, onde se cria uma fórmula sem o uso de dados, com conhecimento prévio do fenômeno físico, premissas ou dinâmica das variáveis envolvidas. A segunda é conhecida por *data-driven*, onde utilizando um conjunto de dados históricos é executado um processo de ajuste dos dados a um modelo.

Neste trabalho, devido a falta de conhecimento prévio das variáveis condicionais para cálculo da probabilidade de *default*, foi desenvolvida a abordagem de modelagem *data-driven*.

A modelagem *data-driven* é estudada no contexto da inteligência artificial, na área de aprendizado de máquina (*machine learning*), e em específico para este trabalho no tema de classificação binária. Este tema também pode ser encontrado na literatura como aprendizado estatístico para estimadores de probabilidade condicional.

Os modelos são ajustados para um conjunto de dados históricos que vincula uma classe binária representada por um valor verdadeiro (1) ou falso (0) e um conjunto de variáveis condicionais.

As variáveis condicionais podem ser chamadas de "variáveis de entrada", "variáveis explicativas" ou *features* e possuem notação X associadas a um vetor ou matriz de variáveis aleatórias.

A variável que representa a classe binária pode ser chamada de "variável de saída", "variável explicada", *label* ou *target* e possui notação Y associada a um vetor de variáveis aleatórias com valores 0 (zero) ou 1 (um).

O processo de ajuste dos dados ao modelo é conhecido como "treinamento", "aprendizado", *learning* ou *fitting*. Uma vez ajustado os dados ao modelo é possível estimar a probabilidade para qualquer novo conjunto de valores das variáveis condicionais.

A variável que representam o resultado da estimativa de probabilidade, é representada com o uso de um acento circunflexo e possue notação $\hat{Y}$ associadas a um vetor

de variáveis aleatórias com valores na faixa de 0 a 1. O uso de um parâmetro de corte T (*threshold*) sobre a variável $\hat{Y}$ possibilita a estimativa da classe com os valores 0 ou 1.

O processo de ajuste aos dados utilizado é o de aprendizado supervisionado, um modelo é ajustado aos dados históricos de entrada e saída utilizando um algorítimo de otimização ou busca. A avaliação de performance do modelo é feita por uma métrica de performance, calculada comparando o valor estimado pelo modelo e o valor real conhecido da variável *target*. A métrica é escolhida de acordo com o problema em estudo, neste trabalho utiliza-se a *ROC AUC* que será explicada na subseção 2.1.1.

Foram utilizados os modelos de classificação binária *XGBoost* e *LightGBM*, que são baseados nos modelos de *Gradient Boosting Trees* (GBT), por sua vez baseados em modelos de árvore de decisão (DT, *decision trees*) e agregação aditiva de novos modelos utilizando o método de *Boosting*[1]. Ambos os modelos utilizados são considerados "estado da arte".

2.1.1 Métrica *ROC AUC*

A função objetivo definida neste trabalho e pela competição no caso prático foi a métrica para classificação binária *ROC AUC*, que pode ser traduzida para "área sobre a curva da característica de operação do receptor". Esta métrica não é diferenciável, e implica que os modelos dependentes de derivadas da função objetivo deverão utilizar uma função de otimização aproximada, também chamada de *proxy*[2], e avaliar a métrica a cada iteração. A literatura de referência para interpretação da métrica *ROC AUC* deste trabalho pode ser obtida em (PRATI, 2008).

Em classificação binária, a variável estimada *target* é frequentemente associada a uma variável aleatória contínua com valores entre 0 e 1 e representada nesta subseção como X. Ela pode ser interpretada como uma pontuação ou uma estimativa de probabilidade condicional calculada para classe binária, a exemplo da estimativa da regressão logística. Dado um parâmetro de corte T, o indivíduo é classificado como positivo (1) se o valor da estimativa X for superior a T e negativo (0) caso contrário. X segue uma densidade

[1] Reamostragem e ponderação das amostras para redução do erro
[2] Função intermediária para aproximar outra função em um problema de otimização

de probabilidade $f_1(x)$ se o indivíduo pertence a classe positiva e $f_0(x)$ caso contrário. A taxa de verdadeiro positivo é dada por T e $f_1(x)$:

$$TPR(T) = \int_T^{\infty} f_1(x)\,dx \tag{1}$$

A taxa de falso positivo é dada por T e f_0x:

$$FPR(T) = \int_T^{\infty} f_0(x)\,dx \tag{2}$$

O formato da curva ROC é determinado pela sobreposição das duas distribuições, e o gráfico indica parametricamente $TPR(T)$ versus $FPR(T)$ variando o parâmetro T para todos os possíveis valores. O incremento de T deve resultar em um número menor de falso positivo (FP) e maior de falso negativos (FN) caracterizado pelo corte vertical do parâmetro T nas curvas f_0 e f_1. Este corte é melhor visualizado na Figura 1.

A seguir é exposto o calculo da $ROC\ AUC$, ou simplesmente AUC, para ilustração:

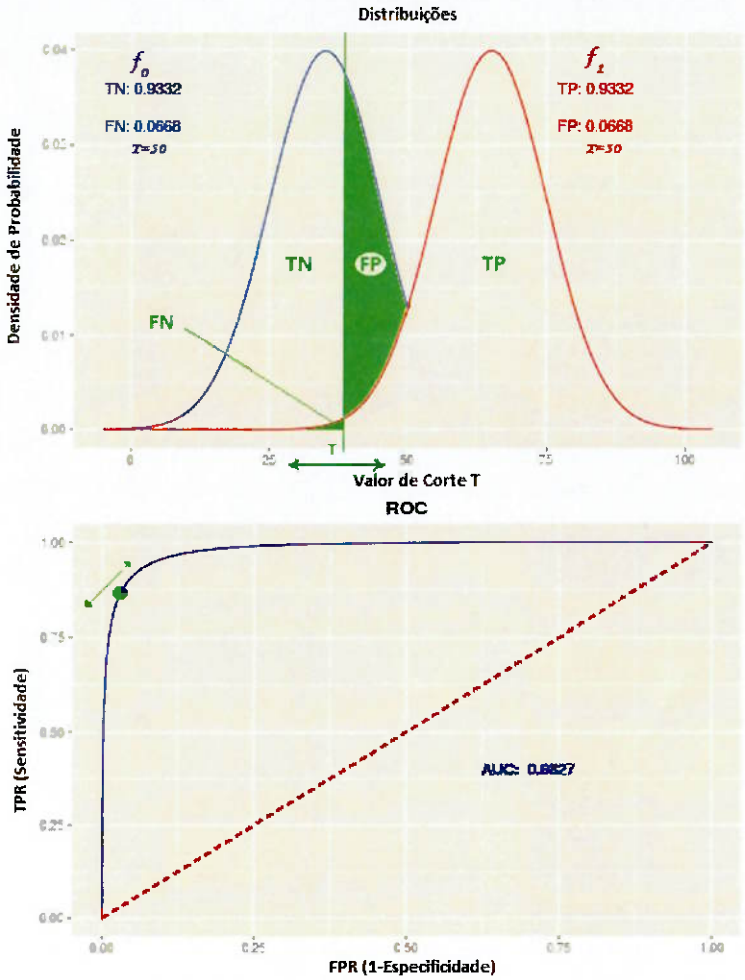
$$\begin{aligned} AUC &= \int_{\infty}^{-\infty} \mathrm{TPR}(T)\mathrm{FPR}'(T)\,dT \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(T' > T) f_1(T') f_0(T)\,dT'\,dT \\ &= P(X_1 > X_0) \end{aligned} \tag{3}$$

Onde FPR' representa $1 - FPR$ ou a especificidade do classificador.

Quando utilizado em unidades normalizadas (*z-score*), a área sob a curva é igual a probabilidade que um classificador irá classificar uma amostra como positiva mais do que uma amostra negativa.

Para exemplificar as possíveis curvas ROC, as densidades associadas f_0 e f_1, o parâmetro de corte T e os conjuntos de verdadeiro positivo (TP), verdadeiro negativo (TN), falso positivo (FP) e falso negativo (FN), nos casos especiais da métrica $ROC\ AUC$ e em um caso usual de diversos classificadores. Serão utilizadas as figuras geradas com auxílio do aplicativo disponível em (RESEARCH, 2016):

Figura 1 – Curva ROC, 0,5 < AUC < 1



A Figura 1 exibe no gráfico superior as densidades de probabilidade f_0 e f_1 associadas ao classificador. É possível notar a existência de interseção das curvas f_0, f_1 e o parâmetro de corte T definindo os conjuntos TP, TN, FP e FN.

Para cálculo da *ROC AUC* o parâmetro T é variado em toda extenção do eixo. A curva *ROC* é exibida no gráfico inferior onde o ponto verde representa a plotagem sequencial da esquerda para direita dos pontos para calculo da equação da *ROC AUC* em cada par TPR e FPR. Os valores de TPR sobem monotonicamente ao incremento do valor de FPR.

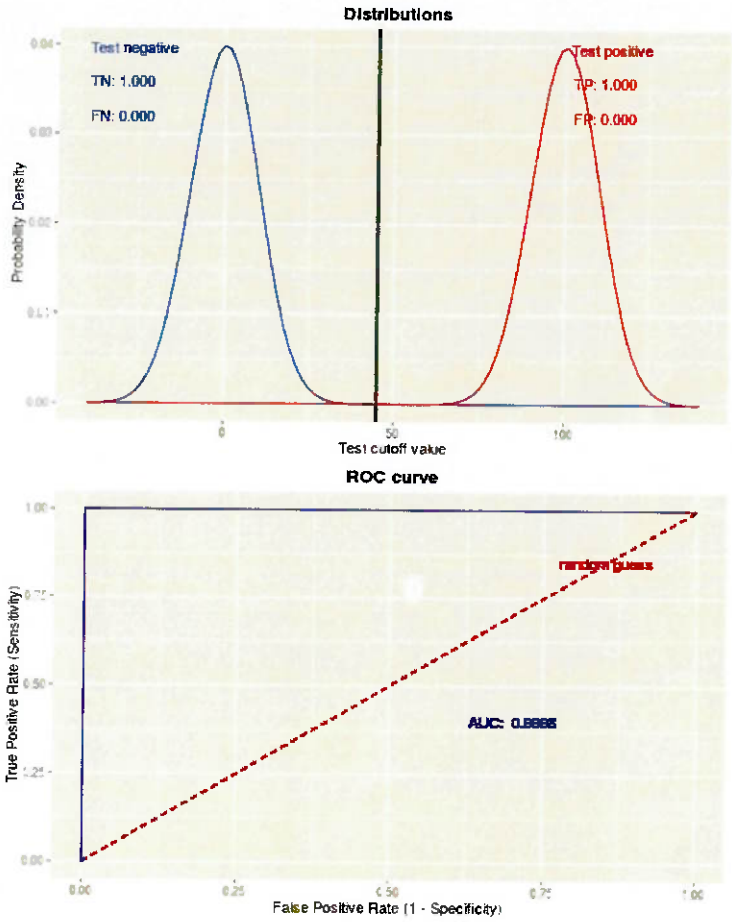
Pelo gráfico desta curva *ROC* pode-se avaliar que existe uma probabilidade de classificação correta superior a 50% nas amostras, independente do valor de corte adotado.

A linha tracejada em vermelho dos pontos (0,0) a (1,1) representa uma curva *ROC* cujo classificador não consegue discriminar as classes positivas e negativas. Este caso

especial será apresentado na Figuras 3. A métrica *ROC AUC* para um classificador que consegue discriminar as classes deve ser superior a 0,5 e igual ou inferior a 1.

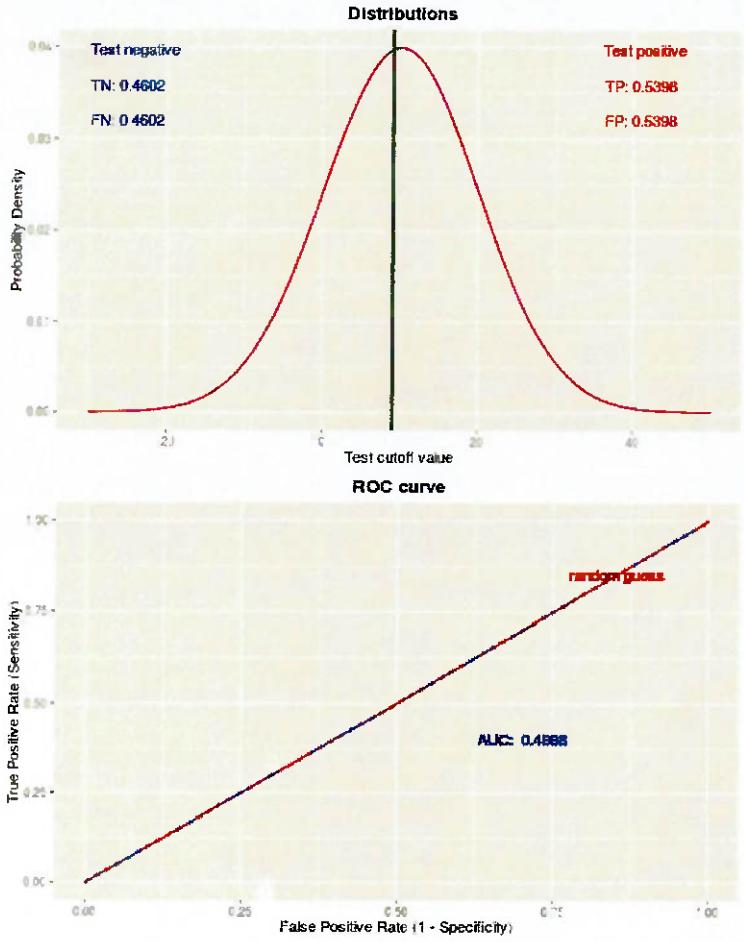
Para os próximos casos as figuras foram obtidas diretamente do aplicativo, sem tradução das legendas e marcação dos conjuntos TP, TN, FP e FN. O parâmetro de corte T foi representado por uma linha vertical na cor preta:

Figura 2 – Curva ROC, AUC = 1



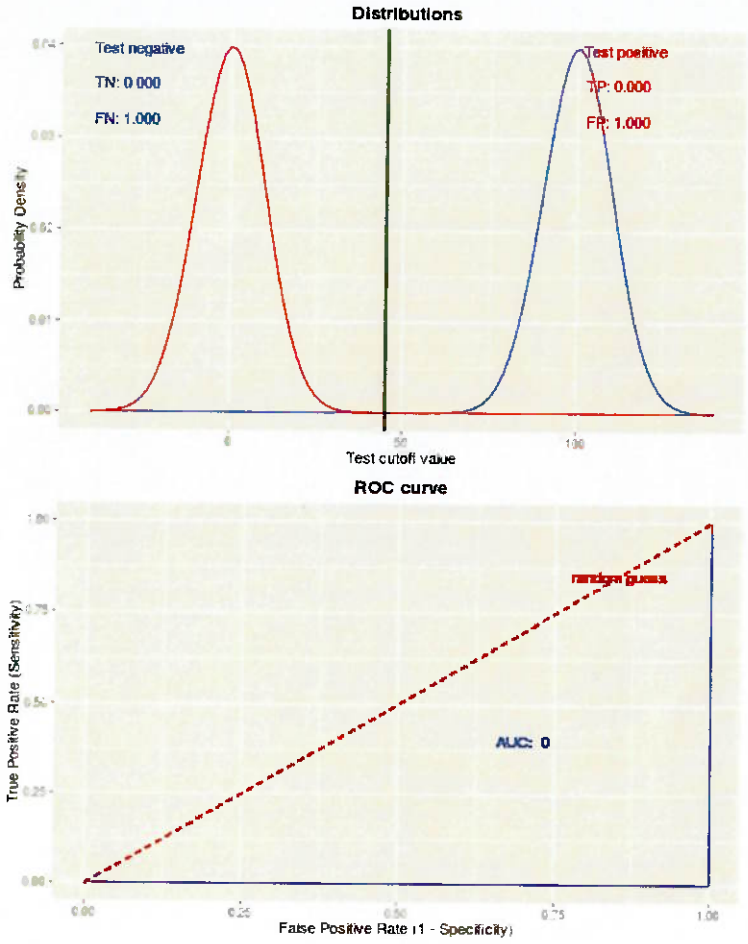
A Figura 2 mostra a representação de um classificador ideal. Os valores para FN e FP são iguais a zero pois o classificador tem a capacidade de discriminação perfeita. A métrica *ROC AUC* apresentada não é exatamente 1, existe uma aproximação numérica para os pontos (0,0) e (1,1) e para as densidades de probabilidade calculadas.

Figura 3 – Curva ROC, AUC = 0.5



A Figura 3 exibe o caso de um classificador incapaz de discriminar as classes 0 e 1. O valor da métrica *ROC AUC* é de aproximadamente 0,5 e as densidades de probabilidade f_0 e f_1 estão sobrepostas.

Figura 4 – Curva ROC, AUC = 0



A Figura 4 exibe o caso de um classificador que discrimina as classes de maneira invertida. O valor da métrica *ROC AUC* é igual a 0.

De forma prática, para métodos de aprendizado supervisionado com uso de otimização e métrica *ROC AUC*, é comum o uso da função objetivo do erro logístico (*logloss*) por ser uma função *proxy* diferenciável para otimização de classificação binária. Neste caso deve-se observar a oscilação da métrica *ROC AUC* a cada iteração da otimização.

2.1.2 Validação Cruzada

O processo de validação cruzada tem por objetivo estimar o erro de generalização de um modelo para um conjunto de dados (amostras) diferente dos dados de treinamento (KOHAVI, 1995).

Um conjunto de dados históricos é dividido em amostras de dados para treinamento e validação. Cada conjunto da divisão (amostra) pode ser chamado de *fold*.

O modelo em avaliação é treinado nos dados de treinamento e avaliado nos dados de validação. A forma de divisão e a combinação das amostras é avaliado de acordo com as características do conjunto de dados. O valor esperado da métrica de performance do modelo sobre os dados de validação representa a estimativa do erro de generalização do modelo, também chamado de "métrica de validação cruzada", "métrica de *cross-validation*" ou "métrica CV". O valor esperado é muitas vezes aproximado pela média aritmética.

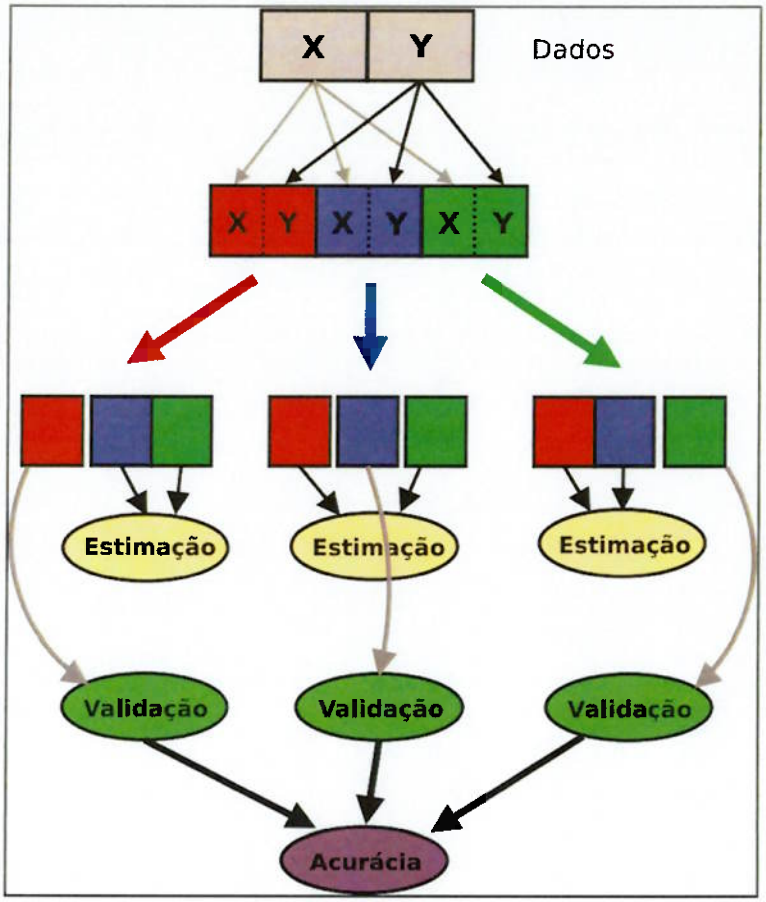
Foi utilizado o método *k-fold* para divisão dos dados. Neste, divide-se o conjunto em k número de *folds* e são realizadas combinações de k *folds* para obter as amostras de treinamento, e 1 (um) *fold* para as amostras de validação a cada combinação. Repetindo-se todas as combinações possíveis de k *folds*, os modelos são ajustados aos dados de treinamento e obtêm-se as métricas de performance nos *folds* de validação.

Para o método *k-fold*, o valor mínimo de k é igual a 2 (dois) e o valor máximo é igual ao número de elementos no conjunto de dados históricos. O método pode receber outra denominação, utilizando o valor máximo ele pode ser denominado como *leave-one-out*, outra denominação conhecida por *leave-p-out* avalia o número de indivíduos nos *folds* de validação onde p representa o número de indivíduos.

As nomenclaturas auxiliam a enfatizar qual divisão foi executada. O processo de validação cruzada é independente ao método de divisão dos *folds*. Boas referências bibliográficas sobre validação cruzada podem ser encontradas em (NETS, 2018) e (HASTIE ROBERT TIBSHIRANI, 2017) onde o objetivo é avaliar qual método se adequa ao conjunto de dados.

A figura 5 ilustra o método *k-fold* com parâmetro k igual a 3 e um processo de validação cruzada com a métrica "acurácia":

Figura 5 – Processo de Validação Cruzada



Fonte: (WIKIPEDIA, 2018)

Na figura 5, o conjunto de dados históricos está representado pelas variáveis X e Y com fundo cinza claro na parte superior ao lado do texto "Dados". As variáveis X e Y representam as variáveis de entrada e saída do modelo e são associadas a cada indivíduo, e não podem ser separadas ou misturadas entre indivíduos. Os conjuntos representados pelas cores vermelha, azul e verde são os *folds* obtidos com o método *k-fold*.

Os grupos de *folds* que foram combinados para criar os conjuntos de treinamento são representados pelas elipses "Estimação" na figura em amarelo claro. Os *folds* de validação associados a cada combinação são representados pelas elipses "Validação" na figura em verde claro.

A cada combinação de *folds*, o modelo é ajustado ao conjunto de dados de "Estimação" e a métrica de performance de validação é obtida nos respectivos dados de "Validação".

A métrica de generalização do modelo, ou métrica da validação cruzada, está representada pela elipse "Acurácia" na figura em cinza escuro.

A métrica de validação individual de cada *fold* pode ser utilizada para avaliar se existe um *fold* com dados "não homogêneos", que pode caracterizar uma variância alta das métricas de validação individuais em relação a outros ajustes de modelos. Esta característica é interessante, um *fold* com métrica divergente da média pode indicar um conjunto de indivíduos *outliers* com características únicas, podendo ser fonte de *insight* para análise.

Cada ajuste de modelo é representado pelo conjunto de dados de treinamento e pelos parâmetros de ajuste de complexidade do modelo, também chamados de "hiper parâmetros" ou simplesmente "parâmetros do modelo". Para um ajuste ideal do modelo por aprendizado supervisionado, deve-se considerar o *tradeoff* de *bias-variance* (RAJNARAYAN, 2018). Em resumo, o *tradeoff* é a escolha dos parâmetros de ajuste ótimo do modelo. Uma variância baixa e uma média alta é obtida nos *folds* de validação, e uma métrica de validação ótima é obtida entre os possíveis valores dos parâmetros do modelo.

O *tradeoff bias-variance* pode ser solucionado por uma metodologia de otimização, onde, as variáveis a serem otimizadas são os "hiper parâmetros" e a função objetivo é a maximização ou minimização da métrica de validação cruzada.

Em modelos com aprendizado por método de otimização, também é utilizado para cada conjunto de "hiper parâmetros" um número de iterações de treinamento com critério de parada ótimo conhecido por *early stop*. O *early stop* tem por função objetivo a maximização ou minimização da métrica de validação cruzada apenas para o conjunto de parâmetros do ajuste aos dados sendo executado.

Os pontos de ótimo obtidos são em geral de máximos ou mínimos locais, e os problemas de otimização são do tipo não convexos com soluções não triviais e obtidas apenas por métodos numéricos.

Conforme (KOHAVI, 1995) em um conjunto de dados históricos grande, o processo de validação cruzada com método *k-fold* e parâmetro k igual a 10 (dez) provê capacidade explicativa comparável ao mesmo método com valor de parâmetro k superior.

Devido ao recurso computacional necessário para executar a validação cruzada, demanda-se tempo avaliando os modelos. O valor do parâmetro k que é suficiente para uma correta avaliação dos modelos, evita o desperdício de recursos, especialmente nas competições e em grandes bases de dados.

O uso do processo de validação cruzada serve também para: seleção do melhor modelo em um conjunto de modelos, agregação de modelos diversos por método de *stacking*[3] e avaliação das variáveis relevantes ao modelo.

A divisão dos *folds* deve ser executada com conhecimento da características dos dados históricos e da forma que eles foram obtidos.

Caso os indivíduos sejam *I.I.D.*[4], a divisão pode ser feita de maneira aleatória em um número previamente estabelecido de *folds* com quantidade de indivíduos igual ou próxima, evitando a duplicidade de indivíduos nos *folds.*

Em casos onde os dados não são *I.I.D.*, pode-se utilizar outras técnicas de divisão como o *walk-forward* para séries temporais, ou método de estratificação dos dados chamado de *stratified k-fold.*

Outra possível solução para dados que não são séries temporais e não se tem a informação que os dados são *I.I.D.*, é a avaliação utilizando ambos métodos de *k-fold* e *stratified k-fold.* Em caso de divergência significante entre as duas métricas, deve-se optar pelo método de *stratified k-fold.*

Em alguns modelos de classificação binária onde o conjunto de dados contém um número maior de indivíduos para determinado valor da variável *target*, existe a possibilidade de viés pelo efeito conhecido como "desbalanceamento de classes". Uma solução é o uso de modelos que tratam o desbalanceamento por ponderação da métrica de performance. Outra solução é o uso de técnicas como a *SMOTE*[5], que fazem uma sub ou sobre amostragem dos dados, podendo em alguns casos duplicar ou remover indivíduos nos *folds.* No uso de *SMOTE*, uma métrica de distância entre os indivíduos pode ser utilizada para definir quais serão removidos ou duplicados.

Em séries temporais utilizando modelos de estimativa futura, deve-se avaliar se o conjunto de treinamento está em um momento do tempo à frente em relação ao conjunto de validação. A autocorrelação dos dados de treinamento contamina os dados de validação, e a autocorrelação futura pode explicar parte das amostras de validação.

De maneira geral, a escolha do método de divisão dos *folds* impacta diretamente na avaliação da validação. Uma técnica incorreta pode gerar viés na escolha das amostras e perda de avaliação da capacidade de generalização do modelo.

[3] Agregação de modelos com uso de folds para geração de meta variáveis e uso de meta modelos para agregação em níveis

[4] Independentes e identicamente distribuídos

[5] *Synthetic Minority Over-sampling Technique*

2.1.3 Elementos de Aprendizado Supervisionado

Aprendizado supervisionado compreende o processo de otimização ou busca para criar modelos que conseguem mapear valores de entrada e saída para um conjunto de dados.

A árvore de decisão é um exemplo de modelo que pode ser utilizado para problemas de aprendizado supervisionado. As *features* X são utilizadas para prever um *target* Y em uma estrutura de árvore com nós de decisão e folhas, ou descrito por alguns autores como grafos.

Modelo e Parâmetros

O modelo em aprendizagem supervisionada se refere à estrutura matemática pela qual a estimativa $\hat{y}_i$ é feita a partir dos dados de entrada X_{ij} e do ajuste prévio do modelo aos dados de entrada X e saída Y. Um exemplo comum é o modelo linear, onde a predição $\hat{y}_i$ é dada como uma combinação linear de variáveis de entrada x_{ij} ponderadas pelos parâmetros (θ_j):

$$\hat{y}_i = \sum_j \theta_j x_{ij} \tag{4}$$

O valor de predição pode ter diferentes interpretações dependendo da tarefa, ou seja, regressão, classificação ou ranqueamento. Em um caso de classificação pode-se transformar "logisticamente" a variável y para obter uma probabilidade, como é o caso do problema de estimação de probabilidade de *default*.

Os parâmetros são os componentes indeterminados do modelo que são obtidos pelo ajuste aos dados históricos, também conhecido por "treinamento" ou "*fitting*". Em problemas de regressão linear, os parâmetros são os coeficientes θ. Em geral se utiliza θ para denominar os parâmetros ajustados dos modelos.

Na modelagem *model-driven*, fórmulas são previamente conhecidas e podem ser estendidas para agregar novos parâmetros ou distribuições de variáveis diferentes. Na modelagem *data-driven*, o número de parâmetros e a estrutura do modelo são desconhecidos, podendo as distribuições de variáveis serem inadequadas para uso nos modelos e sendo

necessário transformações das variáveis. Em alguns casos vários modelos e parâmetros são testados e avaliados para se obter um modelo final.

O modelo pode ser classificado conforme sua complexidade de interpretação. Os modelos são classificados como *black box* quando são complexos, como o caso de redes neurais profundas (*Deep Learning*). Modelos simples com poucos parâmetros, estruturas lineares pequenas ou árvores de decisão pequenas, tendem a ser facilmente interpretadas e podem receber a classificação *white box*. Modelos simples agregados recebem a classificação *gray box* e podem receber a classificação *black box* quando existe um número grande de agregações de modelos. Esta definição pode ser subjetiva e serve para seleção de modelos. Em geral, os modelos *white box* e *gray box* são preferidos pela facilidade de interpretação, otimização, uso, velocidade de treinamento, velocidade de estimação e capacidade de execução em grandes bases de dados.

Função objetivo: Erro de treinamento + Regularização

Com escolhas sensatas para y_i pode-se expressar uma variedade de tarefas para um conjunto de dados, como regressão, classificação e ranqueamento. A tarefa de treinar o modelo equivale a encontrar o valor dos parâmetros θ que melhor se ajusta aos dados de treinamento X_i e y_i. Para treinar os modelos baseados em otimização, define-se a função objetivo para maximização ou minimização adequando-se os parâmetros e estruturas do modelo aos dados de treinamento.

Uma característica marcante das funções objetivo é que elas consistem em duas partes: erro de treinamento ou métrica de performance, e termo de regularização. Estas funções objetivo podem ser escritas como:

$$obj(\theta) = L(\theta) + \Omega(\theta) \tag{5}$$

Onde L é a função de erro de treinamento, Ω é o termo de regularização e θ o conjunto de valores e estruturas de parâmetros do modelo para um conjunto de dados. O erro de treinamento mede o erro associado a estimação do modelo em relação aos dados de

treinamento. Uma escolha comum de L é a função de erro quadrático médio, que é dada por:

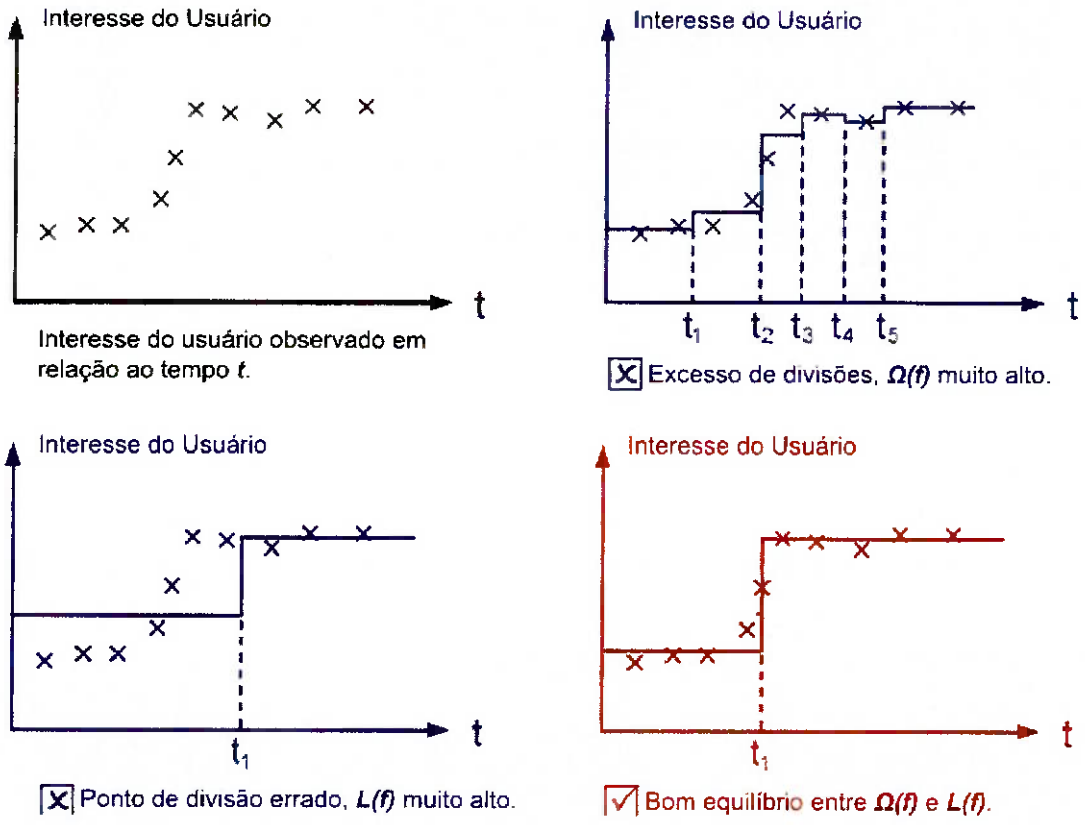
$$L(\theta) = \sum_i (y_i - \hat{y}_i)^2 \tag{6}$$

Outra função de erro comum é o erro logístico, utilizada para regressão logística, estimativa de probabilidade, função *proxy* para otimização ou problemas de classificação binária. Ela é definida como:

$$L(\theta) = \sum_i [y_i \ln(1 + e^{-\hat{y}_i}) + (1 - y_i) \ln(1 + e^{\hat{y}_i})] \tag{7}$$

O termo de regularização auxilia no controle da complexidade do modelo e pode evitar sub e sobre otimização (*undefitting* e *overfitting*). Como isso soa um pouco abstrato, usamos o problema na Figura 6 como exemplo:

Figura 6 – Ajuste função degrau



Fonte: (XGBOOST, 2016)

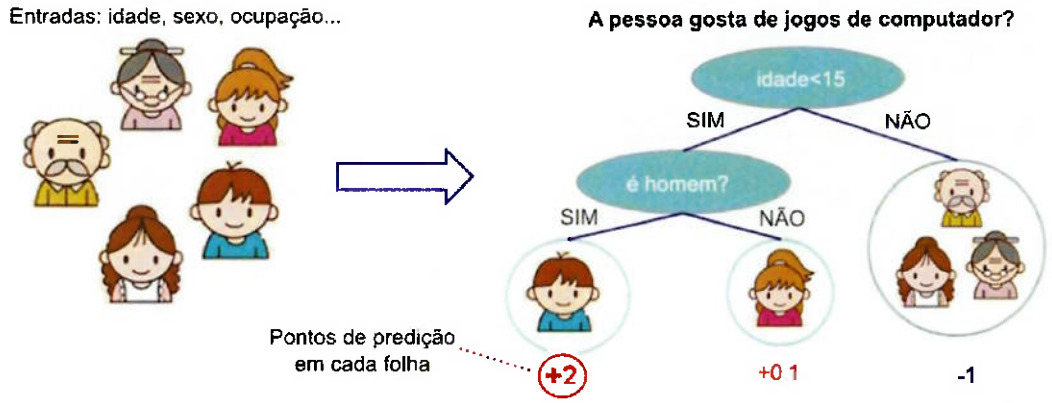
Na Figura 6, deve-se ajustar uma função degrau obtida por amostragem e representada como os pontos "x". A melhor escolha é uma função que tenha boa relação

entre erro e complexidade. A resposta correta está marcada em vermelho. Visualmente parece um ajuste razoável com poucos parâmetros para divisão dos dados no eixo t e erro relativamente pequeno entre o valor dos pontos "x" e a estimaçãodo modelo representada pela linha. O princípio geral é obter um modelo simples e preditivo que se ajuste de forma generalista e não especialista aos dados.

Agregação de árvores de decisão

Para este tópico foi utilizado o modelo *XGBoost*, um modelo de conjuntos de árvores de decisão com utilização de otimização por gradiente e agregação com técnica de amostragem *boosting*. O modelo consiste em um conjunto de árvores de classificação e regressão (CART) agregados. Na figura 7 tem-se uma única CART que classifica se alguém gosta de jogos de computador:

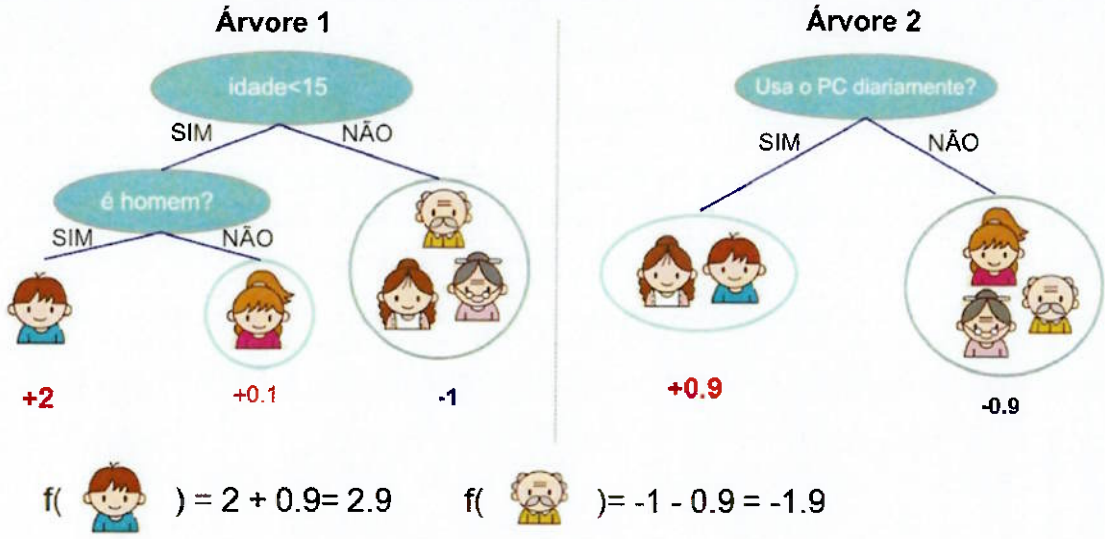
Figura 7 – Exemplo de árvore decisão



Fonte: (XGBOOST, 2016)

No exemplo, classificam-se os membros de uma família em folhas diferentes e é atribuído a pontuação na folha correspondente. Diferente das árvores de decisão padrão, em que a folha contém apenas valores de decisão verdadeiros ou falsos, na CART uma pontuação real é associada a cada uma das folhas, o que gera interpretações mais ricas que vão além da classificação binária, isso permite uma abordagem unificada e baseada em princípios de otimização para criação de modelos de árvores. Em geral uma única árvore não é suficiente para ser utilizada, o que é utilizado é o modelo agregado de árvores que somam as previsões de várias árvores, como no exemplo da figura 8:

Figura 8 – Agregação de 2 árvores



Fonte: (XGBOOST, 2016)

No exemplo, a agregação de modelos com um conjunto de duas árvores. As pontuações de previsão de cada árvore são somadas para obter a pontuação final, sendo possível que árvores com variáveis diferentes se complementem. Matematicamente, pode-se escrever o modelo na forma:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F} \tag{8}$$

Onde K é o número de árvores, f é uma função no espaço funcional F, e F é o conjunto de todos os CARTs possíveis. A função objetivo a ser otimizado é dada por:

$$obj(\theta) = \sum_{i}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{9}$$

Onde l representa a função erro entre a variável conhecida y_i e a variável estimada $\hat{y}_i$ e $\Omega(f_k)$ representa a regularização associada a árvore f_k.

Modelos de agregação de árvores conhecidos como "florestas aleatórias" (*random forests*, RF) e "árvores impulsionadas" (*boosted trees*) são os mesmos modelos, a diferença surge de como é executado o ajuste aos dados (treinamento), uma árvore é treinada e sucessivamente novas árvores são agregadas para maximizar a função objetivo. Em *random forests* árvores são criadas em amostras aleatórias. Em *boosted trees* árvores são criadas nas amostras aleatórias com pesos diferenciados para os indivíduos que obtiveram erros maiores, com o objetivo de diminuir o erro dessas amostras nas árvores subsequentes.

Árvore impulsionada - *Tree Boosting*

Ao observar uma árvore de decisão e comparar com um modelo de regressão linear, a dúvida que surge é: quais são os parâmetros das árvores de decisão? É necessário aprender quais são as funções f_i que contém a estrutura de nós e resultados das folhas da árvore. A estrutura de árvore é mais complexa do que o problema de otimização tradicional de funções diferenciáveis no qual é possível otimizar o objetivo via gradiente. A criação da estrutura das árvores individuais não é tratada nesta explicação, alguns algoritmos de criação de árvores conhecidos são CART, C4.5 e CHAID. *XGBoost* e *LightGBM* implementam um algoritmo próximo ao CART.

É computacionalmente intratável aprender todas as árvores de uma única vez. Assim, utiliza-se uma estratégia aditiva: corrigir o que foi aprendido e adicionar uma nova árvore. Escreve-se o valor de predição na etapa t do modelo de adição, também conhecido como *round* ou iteração, como $\hat{y}_i^{(t)}$:

$$\begin{aligned}
\hat{y}_i^{(0)} &= 0 \\
\hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\
\hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\
&\dots \\
\hat{y}_i^{(t)} &= \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)
\end{aligned} \tag{10}$$

Resta saber qual árvore (f_i) utilizar em cada etapa, o natural é adicionar aquela que melhor otimiza a função objetivo.

$$\begin{aligned}
obj^{(t)} &= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^{t} \Omega(f_i) \\
&= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + \text{constant}
\end{aligned} \tag{11}$$

Reescrevendo e considerando o uso do erro quadrático médio (MSE) como nossa função de erro, o objetivo se torna:

$$\begin{aligned} obj^{(t)} &= \sum_{i=1}^{n}(y_i - (\hat{y}_i^{(t-1)} + f_t(x_i)))^2 + \sum_{i=1}^{t}\Omega(f_i) \\ &= \sum_{i=1}^{n}[2(\hat{y}_i^{(t-1)} - y_i)f_t(x_i) + f_t(x_i)^2] + \Omega(f_t) + \text{constant} \end{aligned} \tag{12}$$

A forma do MSE é de fácil visualização, com um termo de primeira ordem (geralmente chamado residual) e um termo quadrático. Para outras funções de erro de interesse (por exemplo, erro logístico), não é tão fácil obter uma forma simples. Então, no caso geral, toma-se a expansão de Taylor da função de erro até a segunda ordem:

$$obj^{(t)} = \sum_{i=1}^{n}[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)] + \Omega(f_t) + \text{constant} \tag{13}$$

Onde g_i e h_i são definidos como:

$$\begin{aligned} g_i &= \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) \\ h_i &= \partial^2_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) \end{aligned} \tag{14}$$

Removendo todas as constantes, o objetivo específico na etapa t torna-se:

$$\sum_{i=1}^{n}[g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)] + \Omega(f_t) \tag{15}$$

Esta equação é a função objetivo de otimização para a nova árvore. Uma vantagem importante desta definição é que o valor da função objetivo depende apenas de g_i e h_i. É assim que o *XGBoost* e modelos GBT suportam funções de erro personalizadas. Pode-se otimizar inúmeras funções, incluindo a regressão logística, usando exatamente o mesmo *solver* que é utilizado com o g_i e o h_i como entrada. Deve-se observar que a função objetivo deve ser diferenciável ou ser possível obter via expansão de Taylor os valores do gradiente e da hessiana, aproximados ou em forma fechada.

Com a função objetivo conhecida não se deve esquecer do termo de regularização, é necessário definir a complexidade da árvore $\Omega(f)$. Redefinindo $f(x)$ como:

$$f_t(x) = w_{q(x)}, w \in R^T, q : R^d \to \{1, 2, \cdots, T\} \tag{16}$$

Onde w é o vetor de pontuação nas folhas, q é uma função que atribui cada conjunto de dados à folha correspondente e T é o número de folhas. No *XGBoost*, define-se a complexidade como:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{17}$$

O termo $\sum_{j=1}^{T} w_j^2$ pode ser interpretado como a norma L2. Pode-se também utilizar na complexidade a norma L1 com o coeficiente *alpha* e representada como $\alpha \sum_{j=1}^{T} w_j$.

Naturalmente, há mais de uma maneira de definir a complexidade, mas a norma L2 utilizada com a função de erro funcionam bem na prática. A regularização é uma parte que a maioria dos modelos de árvores trata com menos cuidado, ou simplesmente ignora. Isso porque o tratamento tradicional da aprendizagem de árvores enfatizava apenas a melhoria da impureza, enquanto o controle da complexidade é deixado para a heurística. Ao defini-lo formalmente, pode-se ter uma ideia melhor do que está sendo "aprendido" e obtém-se modelos de bom desempenho prático. Depois de reformular o modelo de árvore, pode-se escrever o valor da função objetivo na t-ésima árvore como:

$$\begin{aligned} obj^{(t)} &\approx \sum_{i=1}^{n} [g_i w_{q(x_i)} + \frac{1}{2} h_i w_{q(x_i)}^2] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \\ &= \sum_{j=1}^{T} [(\sum_{i \in I_j} g_i) w_j + \frac{1}{2}(\sum_{i \in I_j} h_i + \lambda) w_j^2] + \gamma T \end{aligned} \tag{18}$$

Onde $I_j = \{i | q(x_i) = j\}$ é o conjunto de índices de pontos de dados atribuídos à j-ésima folha. Observando-se que na segunda linha altera-se o índice da soma porque todos os pontos de dados na mesma folha obtêm a mesma pontuação. Pode-se comprimir ainda mais a expressão:

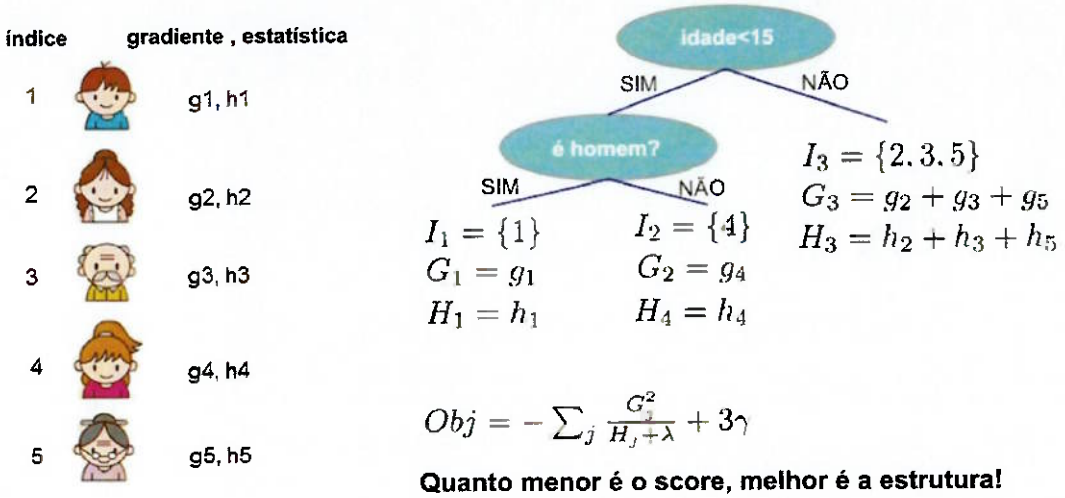
$$\begin{aligned} obj^{(t)} &= \sum_{j=1}^{T} [G_j w_j + \frac{1}{2}(H_j + \lambda) w_j^2] + \gamma T \\ G_j &= \sum_{i \in I_j} g_i \\ H_j &= \sum_{i \in I_j} h_i \end{aligned} \tag{19}$$

Nesta equação, w_j são independentes, a forma $G_j w_j + \frac{1}{2}(H_j + \lambda)w_j^2$ é quadrática, e o valor que otimiza w_j para uma dada estrutura $q(x)$ e a função objetivo é:

$$
\begin{aligned}
w_j^* &= -\frac{G_j}{H_j + \lambda} \\
\text{obj}^* &= -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j + \lambda} + \gamma T
\end{aligned}
\tag{20}
$$

A última equação mede quão boa é uma estrutura de árvore $q(x)$, a exemplo:

Figura 9 – Função objetivo, qualidade/pontuação da árvore



Fonte: (XGBOOST, 2016)

Basicamente, para uma dada estrutura de árvore, "herda-se" as estatísticas g_i e h_i para as folhas, soma-se as estatísticas e calcula-se quão boa é a árvore. Essa pontuação é como a medida de impureza em uma árvore de decisão padrão, exceto que também leva em conta a complexidade do modelo.

Toda a teoria apresentada acima foi desenvolvida e implementada no modelo *XGBoost* e pode ser verificada em (XGBOOST, 2016). Os textos relativos ao modelo *LightGBM* podem ser encontrados em (MICROSOFT, 2018).

A explicação é de interesse prático, pois os modelos *GBT* utilizados neste trabalho devem ser capazes de utilizar a métrica *ROC AUC*.

Diferenças dos modelos *LightGBM* e *XGBoost*

No contexto de modelos GBT, ambos modelos *XGBoost* e *LightGBM* são considerados "estado da arte", e tomaram atenção ao serem utilizados com sucesso em competições e casos reais, com resultado na redução do trabalho para tratamento dos dados e em aumento significativo de performance dos modelos.

Até o momento deste trabalho são relatadas na documentação do *LightGBM* somente duas sutis diferenças entre o *XGBoost*, que serão traduzidas a seguir.

"O *LightGBM* classifica um histograma para variáveis categóricas de acordo com seus valores acumulados ($\sum gradient / \sum hessian$) e, em seguida, encontra a melhor divisão no histograma classificado. O *XGBoost* não faz distinção para variáveis categóricas. Ambos modelos trabalham de forma muito semelhantes porém o resultado final é diversificado, esta diversificação é interessante para uma abordagem de agregação dos modelos. Segundo os autores o processo de criação de árvore também difere em pequenos detalhes."

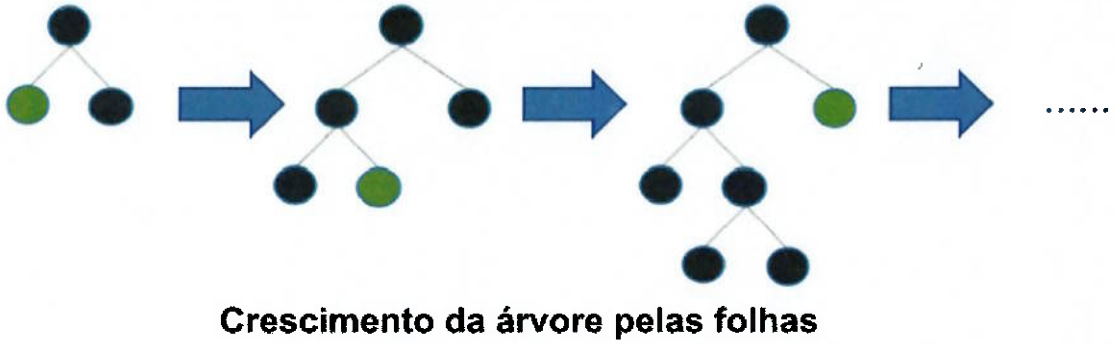
"O *XGBoost* é uma biblioteca de otimização de gradiente distribuída otimizada projetada para ser altamente eficiente, flexível e portátil. Ele implementa algoritmos de aprendizado de máquina sob a estrutura *Gradient Boosting*. O *XGBoost* fornece um reforço de árvore paralela (também conhecido como GBDT, GBM) que resolve muitos problemas de ciência de dados de maneira rápida e precisa. O mesmo código é executado em ambientes distribuídos (Hadoop, SGE, MPI) e pode resolver problemas além de bilhões de indivíduos."

"*LightGBM* aumenta o numero de árvores com novas folhas (*level-wise tree growth*), ele irá escolher a folha com erro máximo de *delta* para crescer. Mantendo as folhas fixas, tende-se a alcançar um erro menor que os algoritmos de crescimento baseado em nível."

Figura 10 – Crescimento da árvore em níveis

Fonte: (MICROSOFT, 2018)

Figura 11 – Crescimento da árvore pelas folha



Fonte: (MICROSOFT, 2018)

Nas Figuras 10 e 11 estão representados as formas de criação de novos nós ou novos níveis nas árvores de decisão.

2.2 Estado da arte no assunto

Para avaliação do conhecimento existente e busca de uma solução próxima ao problema apresentado, foi consultado os seguintes sites de *papers* acadêmicos: (GOOGLE, 2018), (ELSEVIER, 2018) e (SSRN, 2018). Foram utilizadas as palavras-chave: "default risk home credit", "default probability home credit", "default risk analysis", "home credit risk". Os primeiros resultados retornados não trataram especificamente o tema de "estimação de probabilidade de *default* com aplicação de modelos de aprendizado de máquina em empréstimos imobiliários". Relaxando um pouco os critérios de pesquisa e removendo a

restrição de artigos com clientes "desbancarizados" é possível avaliar diversos artigos que estão relacionados com imóveis e empreśtimos.

O trabalho mais relevante observado no assunto foi o (LAWRENCE L. DOUGLAS SMITH, 1992) que trata do problema de análise de *default* para empréstimos imobiliários. Abaixo iremos destacar os pontos principais do trabalho.

No trabalho foi utilizado ferramenta de regressão logística em dados de transação, termos de empréstimos, condições econômicas e outras características. A solução proposta não faz referência ao uso da validação cruzada e modelos de árvore de decisão. O volume de dados avaliado possui um total de 170.000 (cento e setenta mil) empréstimos de uma base de dados monitorada ativamente, fornecendo confiabilidade nos dados e possibilitando a inclusão de fatores externos ao modelo.

Observa-se que o problema de estimação de *default* é estudado a vários anos, com referências para trabalhos do ano de 1969 (mil novecentos e sessenta e nove). Alguns relatos sobre os problemas reais e importância do tema são feitos, como: "sendo possível o acesso aos riscos (a estimativa dos riscos), é possível aumentar a eficiência do mercado de hipotecas pelo aperfeiçoamento da precificação, configurações dos termos, e outras técnicas de alocação." e "a inabilidade para diagnosticar estes riscos pode resultar em perda de oportunidade de lucros, perdas em empréstimos, e praticas de minimização de risco como as '*red lining*'".

Foi relatado que "existe uma falta de base de dados públicas disponíveis para pesquisa". O trabalho forneceu algumas ideias de avaliação das variáveis estudadas, algumas referências informam que a variável de maior capacidade de discriminação do *default* é a taxa do valor do empréstimo e bem financiado, a mesma observação é feita na conclusão neste trabalho.

Foi observado o uso de palavra-chave como "mortgage literature" e "defaulted mortgages" para relatar alguns trabalhos relacionados. Uma revisão sobre *default* de empréstimos foi apresentada, relacionando estudos que avaliam os comportamentos de clientes que incorreram em *default* e não os valores ou variáveis do conjunto de dados. Esta explicação é importante pois relata a dinâmica do fenômeno físico que é retratado pelas variáveis do modelo.

Conforme observado nas primeiras consultas de literaturas, este trabalho relata que: a estimativa de risco global de crédito da instituição é bem conhecida e estabelecida, pode-se observar pela quantidade de trabalhos na área, porém o risco particular de empréstimo

envolve uma dificuldade maior de avaliação e este tem uma maior importância para as empresas. A conclusão do trabalho aborda a informação sobre as variáveis relevantes e as respectivas frequências para amostragem de dados temporais.

Em um segundo trabalho (ADDO DOMINIQUE GUEGAN, 2018) foi relatado o uso de modelos mais atuais, como *random forest* e *deep learning*. O trabalho relata a análise do risco de crédito de empresas, com avaliação de variáveis para mapear o possível risco envolvido. Este trabalho se difere do (LAWRENCE L. DOUGLAS SMITH, 1992) por utilizar técnica de validação cruzada na avaliação do modelo, uso de modelos mais atuais e dados de crédito relativo a empresas e não empréstimos imobiliários.

Comparando os trabalhos, observa-se que (LAWRENCE L. DOUGLAS SMITH, 1992) aborda o problema com um objetivo de explicação das variáveis e dinâmica do fenômeno físico, com referência a trabalhos que abordam o comportamento dos envolvidos e uma boa experiência com negócios imobiliários e empréstimo, realmente é um trabalho que deve ser lido por quem esta conhecendo a área. Enquanto (ADDO DOMINIQUE GUEGAN, 2018) utiliza uma abordagem mais direta do uso de modelos sobre o conjunto de dados, uma abordagem bastante utilizada por cientistas de dados atualmente.

2.3 Contribuição deste trabalho na literatura existente

O trabalho aborda o problema de maneira prática e orientada a modelagem *data-driven*. Devido à característica da metodologia adotada, pode-se obter modelos variados para o problema de risco de crédito em geral, sendo necessário a adaptação para os diferentes conjuntos de dados e avaliação das variáveis relevantes. A base do trabalho é uma estimativa da probabilidade de *default* com uso de modelos de classificação binária por método de aprendizado supervisionado com o uso de validação cruzada nas etapas de modelagem, sendo possível o uso de modelos de árvore de decisão, lineares ou qualquer outro modelo por aprendizado supervisionado.

A diferença para os trabalhos de (LAWRENCE L. DOUGLAS SMITH, 1992) e (ADDO DOMINIQUE GUEGAN, 2018) está na forma de como avaliar novas interações de variáveis. Neste trabalho será utilizada uma abordagem de uso das medidas de relevância de variáveis em cada modelo. A criação das variáveis será executada pela divisão das variáveis, contagem do número de valores nulos e utilização de média aritmética entre

variáveis. A utilização desses métodos pode acarretar na criação das variáveis propostas por (LAWRENCE L. DOUGLAS SMITH, 1992). Não existe nenhuma avaliação relevante a ser feita para o trabalho de (ADDO DOMINIQUE GUEGAN, 2018), neste os pontos principais de modelagem são tratados de forma semelhante. Não foi exposto uma forma de interação de variáveis, porém o uso das redes neurais pode acarretar em um número muito maior de interações mesmo não sendo avaliado a relevância destas relações para o modelo.

É provável que o trabalho não caracterize um novo conhecimento, trata-se da aplicação de um conjunto de técnicas para uma solução geral do problema. Os fatores para esta conclusão são: falta de capacidade em encontrar outros trabalhos relevantes na área específica e livros dedicados para estimação de probabilidade de *default*; existência de trabalhos relativamente antigos na área mas não restrito aos clientes "desbancarizados"; e existência de trabalhos em área de risco de crédito de empresas.

3 Desenvolvimento do modelo / teoria

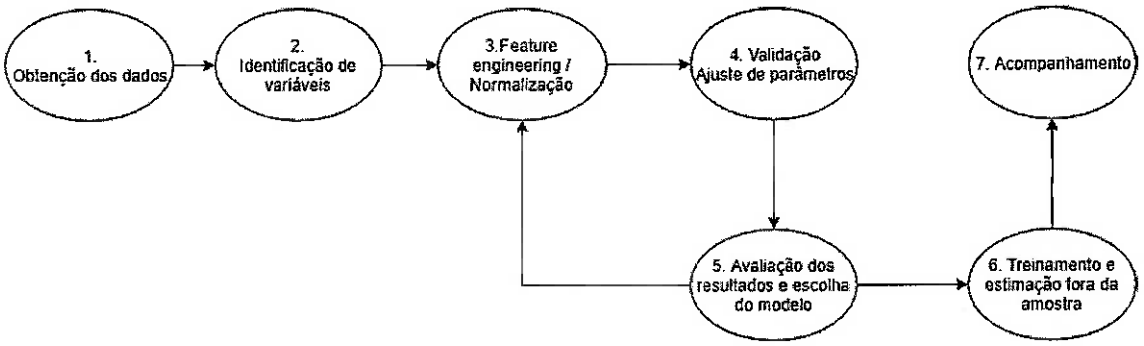
3.1 Aprofundamento da caracterização do estudo

O objeto de estudo é a estimação de probabilidade de *default* com aplicação de modelos de aprendizado de máquina em empréstimos imobiliários, caracterizado por um conjunto de dados históricos que representa a situação financeira, social e econômica de um cliente, um bem financiado no passado, a situação macroeconômica e a informação de *default* referente ao financiamento. Não é objeto de estudo os clientes que foram rejeitados pela empresa de crédito e não incorreram em financiamento e risco de crédito, por não estarem disponíveis no conjunto de dados históricos. Porém a modelagem possibilita a estimação da probabilidade de *default* destes clientes, caso os dados sejam fornecidos.

O processo representado pelo conjunto de etapas de obtenção dos dados até a estimação da probabilidade é chamado de "*pipeline*". Dado a caracterização do problema de classificação binária pode-se utilizar um *pipeline* que divide as etapas de estudo:

1. Obtenção dos dados
2. Identificação e avaliação das variáveis
3. Normalização, criação e transformação de variáveis (*Feature engineering*)
4. Validação do modelo com base na métrica de performance em validação cruzada
5. Avaliação dos resultados de validação e escolha dos modelos
6. Treinamento final para estimar previsões fora da amostra de dados históricos
7. Estimação, uso do modelo e acompanhamento

Figura 12 – Pipeline de criação e avaliação de modelos



3.2 Abordagem do objeto de estudo

É esperado que os dados sejam previamente obtidos para utilizar a abordagem deste trabalho. Esta é feita por processamento de dados, com uso de *software*, e é inviável a execução de forma manual.

Executar todo *pipeline* pode demandar bastante tempo e recurso computacional, deste modo, é importante ter conhecimento de quantas vezes as etapas são geralmente executadas.

As etapas de 3 (três) a 6 (seis) são executadas quantas vezes forem necessárias para se obter novos modelos. As etapas 1 (um) e 2 (dois) são executadas apenas uma única vez. A etapa 7 (sete) é executada de acordo com o limite físico do problema ou objetivo de estudo, que, no caso de uma competição é o número máximo de estimativas por dia e por equipe. É ilimitado o número de variáveis na etapa 3 (três), assim, utilizar uma metodologia de exploração dos dados e restringir o número de interação de variáveis é importante para não exceder o tempo disponível.

A obtenção dos dados nas etapas 1 (um) e 2 (dois) é feita com uso de bibliotecas que facilitam a leitura e criação de *dataframes*. *Dataframes* são análogos as matrizes matemáticas com a possibilidade de utilização de dados não numéricos e utilização de lógica relacional entre diversos *dataframes*.

Na análise dos dados utilizam-se bibliotecas para apresentação de gráficos, e bibliotecas de estatística descritiva para analisar as distribuições das variáveis. O conhecimento da distribuição pode auxiliar na criação de interações entre variáveis.

Para criar os modelos que relacionam as variáveis explicativas (*features*) e a probabilidade de *default*, serão utilizados modelos de árvores de decisão com a capacidade de tratamento dos dados nulos. O uso desses modelos facilita a análise dos dados e remove uma etapa de tratamento ou transformação.

Não será feito uma análise do fenômeno físico do problema na abordagem do objeto de estudo, é esperado no mínimo um conjunto de dados com informações dos empréstimos e os respectivos valores das classes de *default*.

O conhecimento é feito em cada caso prático com os respectivos dados, conhecer o fenômeno físico pode auxiliar na obtenção de melhores modelos de estimação. O conhecimento parcial pode ser obtido por meio da avaliação da relevância das variáveis, conhecendo

os casos onde o *default* é mais provável. A falta do conhecimento do significado de uma variável relevante, pode levar a casos de correlação espúria onde a variável estudada não faz parte do fenômeno físico. Estas afirmações são necessárias pois a cada novo conjunto de dados o conhecimento pode ser modificado.

Com uso de modelos de árvores de decisão na etapa 3 (três), pode-se avaliar quais as variáveis que possivelmente possuem uma capacidade explicativa maior. Isto é feito com a utilização da relevância das variáveis obtidas pelo ajuste do modelo.

Algumas informações que caracterizam a relevância são: medida de ganho de informação de entropia e gini, valores das métricas da otimização gi e hi, ou contagem do número de nós onde cada variável foi utilizada.

A etapa 5 de escolha dos modelos é feita utilizando a estimativa de generalização do modelo por validação cruzada. O modelo com maior métrica é selecionado para a etapa 6 de treinamento do modelo final.

A abordagem finaliza com a estimação de novos conjuntos de dados na etapa 7 (sete) e acompanhamento do modelo. Caso o modelo apresente uma performance ruim nos novos conjuntos de dados, o processo é iniciado pela etapa 1 com a consideração que novos dados podem ser adicionados para melhoria do modelo.

3.3 Modelagem

A modelagem segue os passos do pipeline da seção 3.1, e cada etapa será explicada a seguir.

O conjunto de variáveis é obtido na etapa 1 (um) com dados históricos para treinamento, ou na etapa 7 (sete) com dados em tempo real para estimação pelo modelo. Os dados são catalogados em banco de dados. Podendo ser armazenados de forma estruturada com padronização das variáveis, ou de forma não estruturada sendo necessário uma etapa adicional de interpretação dos dados para extração de dados estruturados.

Um exemplo de dado não estruturado é o texto de uma carta ou contrato, e de dados estruturados uma data, um endereço ou um saldo de conta bancária. É esperado que seja fornecido um resumo do significado de cada variável dos dados, e como os diversos arquivos se relacionam para um mesmo cliente ou financiamento. O conjunto de informações que explicam os dados de forma organizada é conhecido por "dicionário de dados".

A etapa 2 (dois), consiste na identificação, validação e avaliação dos valores que as variáveis podem apresentar e estão apresentando no conjunto de dados. Nesta etapa, caso o dicionário de dados não exista, um dicionário deve ser criado com as observações feitas na avaliação do conjunto de dados. O objetivo é avaliar possíveis erros de coleta de dados e validação dos dados.

Em alguns problemas as relações entre vários conjuntos de dados é feita com uso de algebra relacional, basicamente união de conjuntos identificados por indivíduos ou categorias.

Os problemas de coleta de dados mais comuns são: existência de dados nulos, valores *outliers* ou fora das faixas de valores para a variável, ocorrência de indivíduos ou variáveis duplicadas e variáveis que não estão no dicionário de dados ou não foram encontradas nos dados.

Durante o processo de avaliação das variáveis, são avaliadas as possíveis interações e correlações entre variáveis para uma melhor compreensão do conjunto de dados e complexidade do problema.

As variáveis podem ser classificadas em quantitativas continuas, quantitativas discretas, qualitativas ordinais, qualitativas nominais e estruturas de dados como textos, imagens, vídeos ou outra estrutura complexa.

Para o uso dos *softwares* de modelagem, as variáveis são classificadas em categóricas, números de ponto flutuante, números inteiros, *booleana*[1], textos e outras estruturas de dados complexas.

A identificação correta do tipo de variável no *software* auxilia na otimização do uso de memória e redução do tempo de processamento dos dados. O conhecimento da estrutura dos dados e do fenômeno físico em estudo, guiam e diminuem as possibilidades de pesquisa, evitando casos de correlações espúrias ou dados inconsistentes pelos modelos nas próximas etapas.

É importante que esta etapa seja executada corretamente, para não ser necessário executá-la mais de uma vez.

A etapa 3 (três) tem por objetivo normalizar, criar e transformar as variáveis para a utilização no modelo de aprendizado de máquina.

[1] Variável que aceita dois possíveis valores: verdadeiro ou falso. São armazenadas na menor unidade de armazenamento, o *bit*

Nesta etapa é possível a conversão de valores para faixas normalizadas por média e variância, por mínimo e máximo dos valores das variáveis ou outra transformação necessária. Extrair informações de dados complexos como imagens, vídeos e textos em conjuntos de novas variáveis. Pode-se também efetuar interação das variáveis como a divisão de uma variável por outra.

É possível converter variáveis qualitativas em uma sequência de números inteiros, processo conhecido como *label encode*. As variáveis transformadas por *label encode* também podem ser representadas por um conjunto de várias variáveis numéricas do tipo 0 (zero) ou 1 (um) para cada valor da variável transformada, processo conhecido como *one-hot encode*.

O estudo de todas as possíveis interações entre variáveis e suas transformações não é tratado neste trabalho, e é estudado no contexto de mineração de dados no tema de *feature engineering*.

É possível executar exaustivamente várias técnicas de *feature engineering* para criar novas variáveis. O crescimento do número de variáveis por *feature engineering* pode ser um problema grave de modelagem com a possibilidade de multicolinearidade, ruído excessivo, correlações espúrias e em alguns casos quebra de informação da variável explicada para as variáveis explicativas, conhecido também por vazamento de dados.

Esta etapa é conhecida por ser a mais importante de todo processo de modelagem *data-driven*. Existem modelos que possuem a capacidade de interação das variáveis, como é o caso de *redes neurais* e podem reduzir o tempo gasto nesta etapa. Outras técnicas podem ser executadas como a redução de dimensionalidade com o uso de *PCA* [2] e *LDA* [3].

A etapa 4 (quatro) avalia quanto o modelo consegue extrair de informação dos dados da etapa 3 (três) para fazer as estimativas.

A avaliação é feita utilizando o processo de validação cruzada(KOHAVI, 1995). Na primeira execução desta etapa em todo o *pipeline*, utiliza-se o menor conjunto de dados e conjunto de parâmetros "padrão" do modelo. Esta primeira métrica de performance obtida em validação cruzada é conhecida por *baseline*.

As execuções subsequentes desta etapa, podem ser utilizadas para otimização dos parâmetros do modelo em busca de melhores performances, avaliação das variáveis relevantes e obtenção de *insights* sobre o conjunto de dados.

[2] *Principal Components Analysis*
[3] *Linear Discriminant Analysis*

O processo de obtenção de parâmetros ótimos é o segundo item que demanda maior tempo e em conjunto com a criação de variáveis são os de maior importância do *pipeline*.

Todos ajustes executados no modelo e avaliados por validação cruzada, devem ser registrados para seleção do modelo ótimo.

Na etapa 5 (cinco) avaliam-se todas as métricas de validação cruzada obtidas na etapa 4 (quatro), e é escolhido o melhor modelo em relação a métrica.

Um bom conjunto de variáveis explicativas utilizadas em um modelo com parâmetros não ótimos, pode levar a uma situação de sub ou sobre ajuste, também conhecidos como *underfit* ou *overfit*.

O sub ajuste se caracteriza por não ajustar o modelo suficientemente aos dados, e o sobre ajuste por ajustar excessivamente. Existe um *tradeoff* conhecido como *bias-variance* para tratar esta escolha de melhor parâmetro. Na prática, deve-se escolher entre modelos que obtiveram uma métrica de validação melhor, e uma variância entre *folds* menor.

Para tratar do problema de ajuste e *tradeoff*, recorre-se a algumas técnicas existentes nos modelos como: a utilização de regularização de parâmetros na função objetivo, a escolha do momento de parada ótimo no ajuste aos dados com uso da métrica de validação cruzada conhecida por *early stop*, e em alguns casos avaliação minuciosa das estimativas dos modelos em valores conhecidos ou *outliers*.

A otimização dos parâmetros do modelo é um problema numérico, sem fórmula fechada, e pode ser feita com uso de busca aleatória, faixa de valores de pesquisa (*grid search*), otimizações livres de derivadas e diversas outras técnicas. Neste trabalho foi utilizada busca manual e otimização *bayesiana*[4].

A validação cruzada também pode ser utilizada para avaliar a relevância das variáveis na etapa 3 (três), para isto, adiciona-se a variável e obtém-se a métrica de validação, em seguida remove-se a variável e obtém-se a métrica de validação, a variação significativa da métrica está relacionada com a relevancia da variável.

Nos casos de multicolinearidade, a relevância da variável pode auxiliar na redução da dimensão do problema. Em casos de correlações espúrias é difícil identificar as variáveis que não representam o fenômeno físico sem o dicionário de dados e conhecimento do fenomeno.

[4] Processo de otimização que utiliza um modelo probabilístico multivariado (representado por uma rede Bayesiana) para gerar novas soluções a cada iteração.

As etapas 3 (três) e 5 (cinco) podem ser repetidas para melhorar a métrica de validação na etapa 4 (quatro) quantas vezes forem necessárias. A variação dos parâmetros com valores próximos, auxilia a detectar sub e sobre ajuste do modelo aos dados nos casos onde a métrica oscila de maneira significativa.

Na etapa 6 (seis) de ajuste do modelo final, os modelos selecionados são treinados sobre todo o conjunto de dados, utilizando os parâmetros obtidos na validação.

Com uso de métodos de agregação de modelos como o *stacking*, pode existir a divisão dos dados em *folds* e a realização de novos ajustes dos modelos. A agregação de modelos nesta etapa deve ser executada com uso das etapas 3 (três), 4 (quatro) e 5 (cinco) como se as estimativas dos modelos fossem variáveis explicativas. Deve-se executar com o cuidado de não ocorrer vazamento de informação da variável explicada para as variáveis explicativas geradas pela estimativa dos modelos.

Na etapa 7 (sete) é executada a estimação em novos dados, e aguarda-se a possibilidade de obtenção da métrica de performance sobre estas estimativas.

Em competições este valor é calculado sobre os dados enviados, na "vida real" é necessário aguardar a ocorrência do fenômeno físico. Nesta etapa também se acompanha a métrica de generalização do erro do modelo nos dados reais. Em casos onde o a estimativa e o valor real começam a divergir e ultrapassar um nível pré estipulado, deve-se reiniciar o *pipeline* e avaliar novos modelos e variáveis.

3.4 Obtenção dos resultados

De forma objetiva, a métrica a ser avaliada neste trabalho é a *ROC AUC*, apresentada no capítulo 2. Para cada modelo e conjunto de dados, deve-se obter a métrica em todas validações cruzadas realizadas. Podendo descartar as métricas dos modelos que não foram escolhidos na etapa 6. Na etapa 7 deve-se avaliar se a métrica da validação cruzada do respectivo modelo se mantém em níveis aceitáveis. Algumas explicações e considerações para este procedimento serão feitas nesta seção.

É possível que o modelo final não tenha capacidade de discriminação. Casos especiais da métrica *ROC AUC* são apresentados no capítulo 2. Deve-se evitar os modelos com valores próximos e inferiores a 0,5 da métrica *ROC AUC*. Uma métrica inferior a 0,5 representa um modelo que está estimando a classe de maneira invertida. No caso de falta

de conhecimento do significado das variáveis e uma métrica ruim, nada pode ser feito e o modelo deve ser descartado.

Deve ser obtido um valor de referência para comparar os demais modelos em validação cruzada. Este valor é chamado de *baseline* e deve ser obtido no modelo com parâmetros "padrão" e um conjunto de dados simples sem interação das variáveis. Isto foi explicado na etapa 4 (quarto) e é de grande importância.

Na "vida real" para casos de crédito imobiliário, a métrica das estimativas pode ser obtida somente após meses ou anos.

Em um problema onde o processo de validação cruzada foi realizado corretamente e o conjunto de dados estimados está "em linha" com os dados de validação, o resultado da métrica de validação tende a ser estável nos dados estimados(KOHAVI, 1995). Por estar "em linha", entende-se que os valores das variáveis explicativas do conjunto de dados estimados estão em faixas de valores proximas das variáveis observadas nos dados históricos.

Uma divergência da métrica pode sugerir algumas hipóteses sobre o *pipeline*: o modelo está sub ou sobre otimizado, as variáveis explicativas foram contaminadas de alguma forma com o valor da variável explicada no processo de criação de variáveis, a divisão dos *folds* foi realizada de forma errada como no caso de autocorrelação em séries temporais, o processo de validação não foi efetuado de forma correta para o problema onde existe alguma característica nas amostras que não são *I.I.D.* ou o fenômeno físico sofreu alteração e o conjunto de dados históricos não é suficiente para explicar o novo regime.

A hipótese de novo regime pode ser caracterizada pelo excesso de outliers e estimativas demasiadamente ruins, ou observado com os valores das variáveis explicativas não estarem "em linha" com o conjunto de dados históricos. Outras hipóteses podem ser levantadas e avaliadas, mas em geral estes são os problemas que ocorrem com maior frequência.

Uma métrica de tempo gasto no *pipeline* é fundamental para uma avaliação subjetiva do modelo e restrição de tempo de execução. Criar as variáveis e ajustar os dados aos modelos não pode demandar tempo superior ao que esta disponível no caso prático, com risco do problema ser tratável apenas a nível de simulação. Para este trabalho o tempo total útil é de 3 (três) meses do inicio ao final da competição, oque possibilita uma gama de tentativas de modelagem e interação, sem limite de tempo de execução do *pipeline*.

4 Aplicação da Teoria

4.1 Descrição/ caracterização do caso prático

Neste trabalho foi avaliado um único caso prático, onde foi aplicado a modelagem da seção do capítulo 3.3.

O conjunto de dados utilizado se refere a competição da empresa *Home Credit* na plataforma *Kaggle*, estes dados estão disponível apenas para os competidores. As referências e imagens apresentadas estão disponíveis de forma pública no *site* da plataforma. Caso o leitor tenha interesse nos dados é necessário aceitar as regras da competição.

Todo problema foi tratado com bibliotecas de *software* voltadas para análise de dados e aprendizado de máquina. Foi utilizado a linguagem de programação *Python 3*, escolhida por não ter restrição na competição, e as bibliotecas *Pandas*, *Numpy*, *Scikit Learn*, *XGBoost*, *LightGBM*, *Matplotlib* e *fmfn*. O editor *Jupyter Notebook* foi utilizado pois possibilita editar documentos e adicionar a execução de programas de forma interativa. O computador utilizado é uma máquina Intel I7-7820x com 128GB de memória ram, com capacidade extremamente alta, o consumo máximo de memória atingido foi de 48GB.

Para obter os dados dos arquivos, foram utilizadas as bibliotecas *Pandas* e *Numpy*, que facilitam a leitura e criação de *dataframes*.

Para entender e analisar os dados, foram utilizadas as bibliotecas juntamente com o editor. Com elas é possível avaliar os conjuntos de dados, criar estatísticas descritivas, verificar os valores das variáveis e obter possíveis *insights* para novas interações de variáveis.

Para criar os modelos, foram utilizadas as bibliotecas *XGBoost* e *LightGBM*, ambas bibliotecas têm a capacidade de tratar dados nulos sem necessidade de estimar um valor.

Para validação cruzada, foram utilizadas as funções de validação cruzada fornecidas em cada biblioteca. A divisão dos *folds* foi realizada por *stratified k-fold* com parâmetro k igual a 5, a escolha se mostrou suficiente onde o aumento de k não alterou de maneira significativa a variância das métricas de performance em validação dos *folds*.

Para otimização dos modelos com ajuste de parâmetros, foi utilizada a biblioteca *fmfn* de otimização *bayesiana* com maximização da métrica de validação cruzada.

4.2 Coleta de dados/ informações

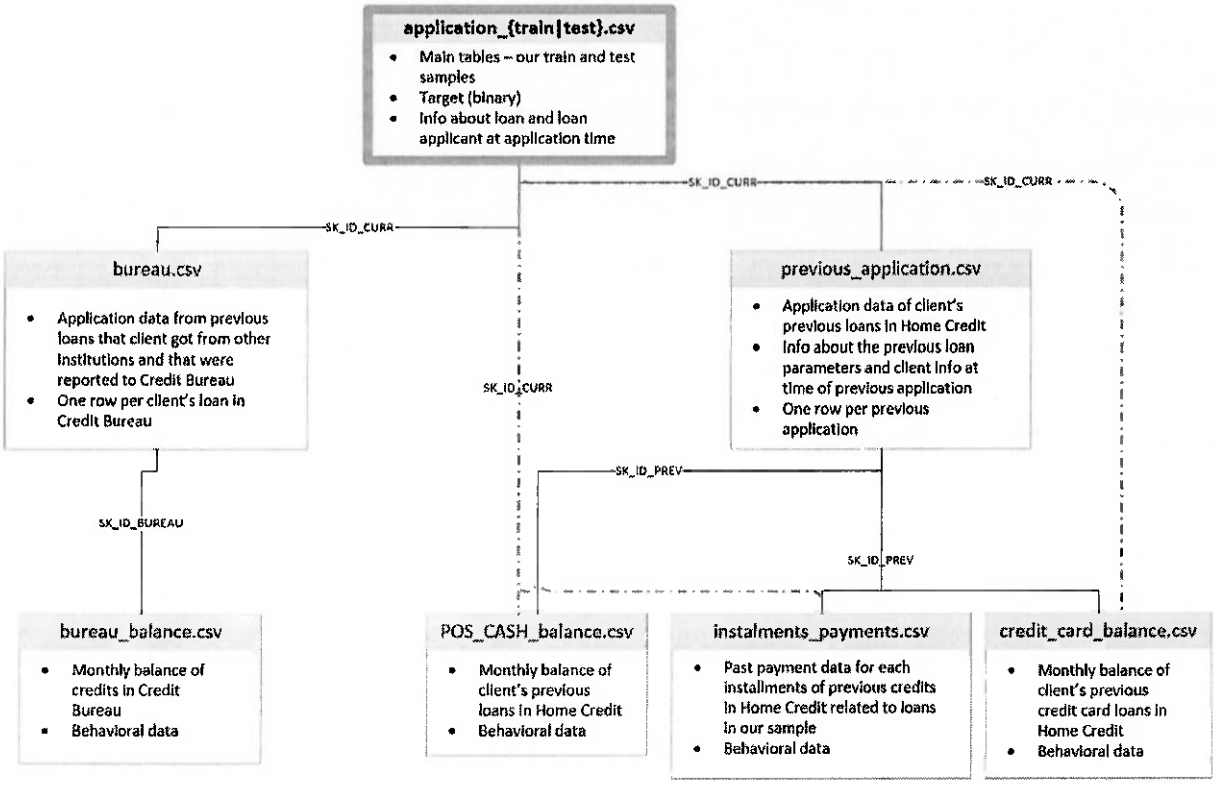
A empresa que formulou a competição entregou juntamente com os dados uma descrição das variáveis, ao total 7 (sete) arquivos de dados:

1. **application_{train|test}.csv** - Tabela principal, dividida em dois arquivos um "Train" utilizado para treinamento/ajuste aos dados, que inclui a informação do *default*, e um arquivo "Test" que serão os dados fora da amostra que deve ser estimada. São dados estáticos para todas possíveis aprovações de crédito. Uma linha representa um empréstimo na amostra de dados.
2. **bureau.csv** - Todos os créditos anteriores do cliente fornecidos por outras instituições financeiras que foram reportados ao Bureau de Crédito (para clientes que possuem um empréstimo na amostra). Para cada empréstimo na amostra, há tantas linhas quanto o número de créditos que o cliente tinha no Bureau de Crédito antes da data de aplicação.
3. **bureau_balance.csv** - Saldos mensais de créditos anteriores no Bureau de Crédito. Esta tabela contém uma linha para cada mês de histórico de todos os créditos anteriores reportados ao Departamento de Crédito.
4. **POS_CASH_balance.csv** - Cópia do balanço mensal dos POS (ponto de vendas) anteriores e empréstimos em dinheiro que o requerente tinha. Esta tabela tem uma linha para cada mês de histórico de cada crédito anterior (crédito ao consumidor e empréstimos em dinheiro) relacionado a empréstimos na amostra.
5. **credit_card_balance.csv** - Cópias de saldo mensal de cartões de crédito anteriores que o solicitante tenha. Esta tabela tem uma linha para cada mês de histórico de cada crédito anterior (crédito ao consumidor e empréstimos em dinheiro) relacionado a empréstimos na amostra.
6. **previous_application.csv** - Todas as solicitações anteriores para empréstimos de crédito residencial de clientes que possuem empréstimos na amostra. Há uma linha para cada solicitação anterior relacionada a empréstimos na amostra de dados.
7. **installments_payments.csv** - Histórico de amortização dos créditos anteriormente desembolsados relativos aos empréstimos na amostra. Existe uma linha para cada pagamento que foi efetuado, mais uma linha para cada pagamento não efetivado.

Uma linha equivale a um pagamento de uma parcela ou uma parcela correspondente a um pagamento de um crédito anterior relacionado a empréstimos na amostra.

Além de cada arquivo, foi reportado uma imagem que explica o relacionamento dos arquivos:

Figura 13 – Relacionamento dos Arquivos



Fonte: (CREDIT, 2018)

Cada arquivo possui uma lista de colunas e seus significados são explicados no arquivo "HomeCredit_columns_description.csv" que é restrito aos competidores. De forma pública é possível visualizar as análises feitas por alguns participantes que retratam a distribuição dos dados.

Para ter idéia da dimensão dos dados, foi exposto de forma publica por um participante a seguinte tabela:

Tabela 1 – Tamanho dos arquivos

Arquivo	Linhas	Colunas
application_train	307511	122
POS_CASH_balance	10001358	8
bureau_balance	27299925	3
previous_application	1670214	37
installments_payments	13605401	8
credit_card_balance	3840312	23
bureau	1716428	17

4.3 Aplicação da teoria ao caso em questão

Pode-se observar pela dimensão dos dados que existem muitas variáveis em diversos arquivos onde alguns arquivos possuem mais de uma relação de linha por solicitação de crédito. Foi reportado também que os arquivos foram fornecidos sem agregação dos dados. O estudo de séries temporais não foi realizado neste trabalho e a solução adotada foi utilizar a primeira linha dos arquivos que continham séries temporais, ou seja, provavelmente o ultimo registro da série. Grande parte dos dados foram descartados pelos modelos com esta decisão.

Além da obtenção dos valores, cabe na primeira etapa a avaliação das variáveis qualitativas e quantitativas. Variáveis qualitativas foram codificadas em números inteiros seguindo a ordenação apresentada no arquivo pelo método *label encode*. O dicionário de dados já informava as colunas qualitativas e foi adotado somente as técnicas de transformação das variáveis. As variáveis quantitativas não foram transformadas. Criação de variáveis foram executadas e serão apresentadas no decorrer da seção.

Após a análise dos tipos de dados em cada variável dos arquivos, foi avaliado a distribuição dos valores nulos, que está apresentado publicamente pelos participantes.

Os valores nulos apresentados são superiores a 50% (cinquenta por cento) dos dados, sendo assim, é de grande auxílio a utilização de modelos que tratam os valores nulos de forma automática. O tratamento por árvores de decisão é feito via comparação direta do valor da variável ser nulo ou não, e é feito automaticamente pelos modelos *XGBoost* e *LightGBM*. Foi criado uma variável por arquivo com o número de valores nulos por indivíduo. Com esta informação é possível inferir um comportamento da obtenção dos dados.

Todas variáveis criadas foram avaliadas por validação cruzada.

Após análise das variáveis com campos nulos, foi observado as distribuições dos valores de todas variáveis. Algumas análises dos dados estão disponíveis publicamente pelos competidores.

Visualizar as variáveis é importante para compreendê-las, encontrar *outliers*, grupos de dados comuns, variáveis correlacionadas de forma não linear e a partir dessas avaliações criar novas variáveis com objetivo de melhorar a performance do modelo. Em alguns casos esta compreensão leva a remover variáveis. Um exemplo de variáveis que foram removidas são as variáveis que disponibilizam informação de média/mediana/moda. Elas estão altamente correlacionadas formando "clusters" na matriz de correlação. A utilização de apenas uma delas foi suficiente para os modelos finais. Também foi testado uma relação direta de divisão pela média, mas não foi relevante para o modelo.

Inicialmente foi obtido a métrica *baseline*, o modelo foi ajustado ao conjunto de dados mais simples possível, executando uma validação cruzada e obtendo a saída das métricas do modelo. Esta primeira avaliação é exibida a seguir com uso do modelo *LightGBM* e sua evolução a cada iteração de treinamento:

Figura 14 – *Early Stop*

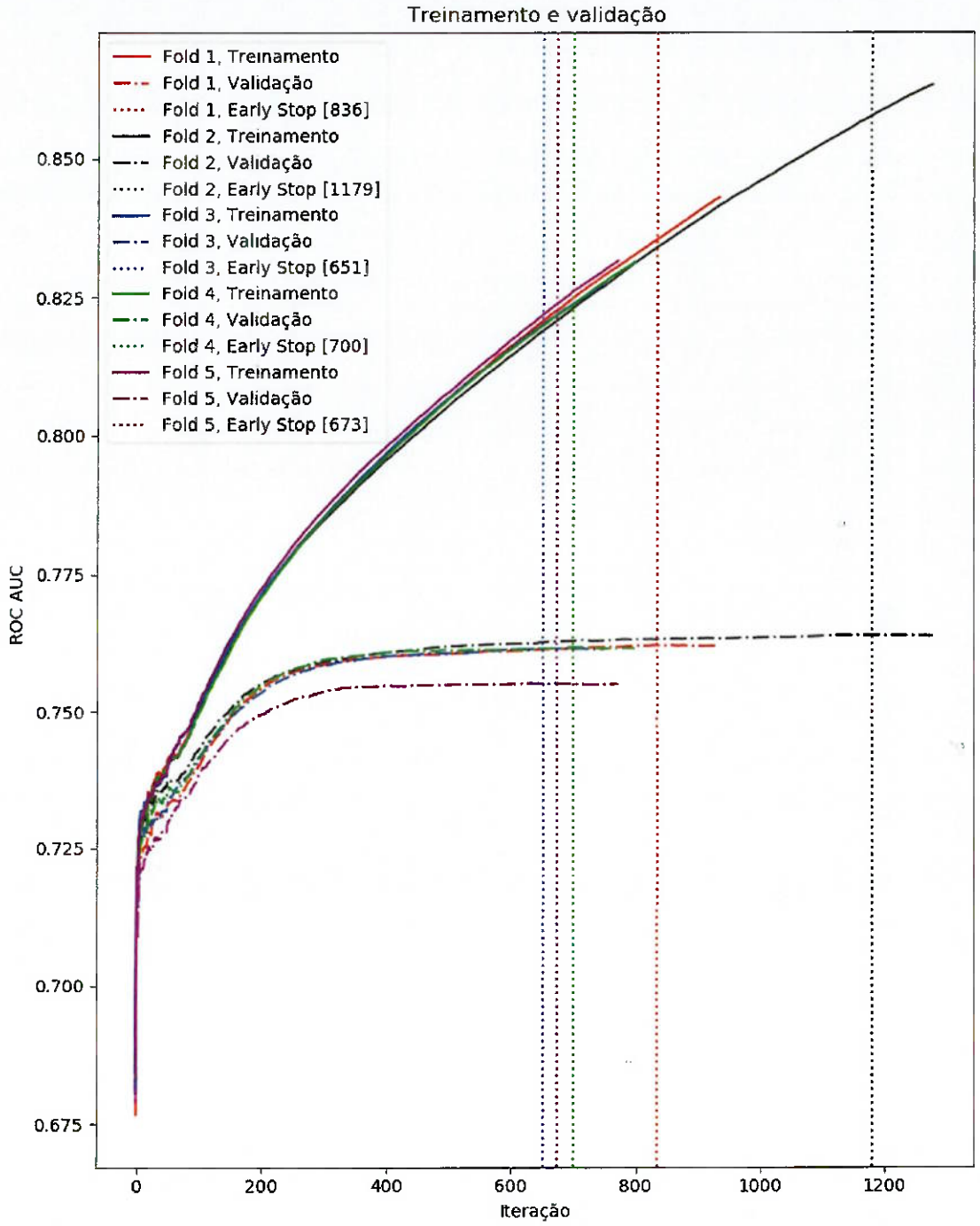


Figura 15 – Curvas ROC por fold

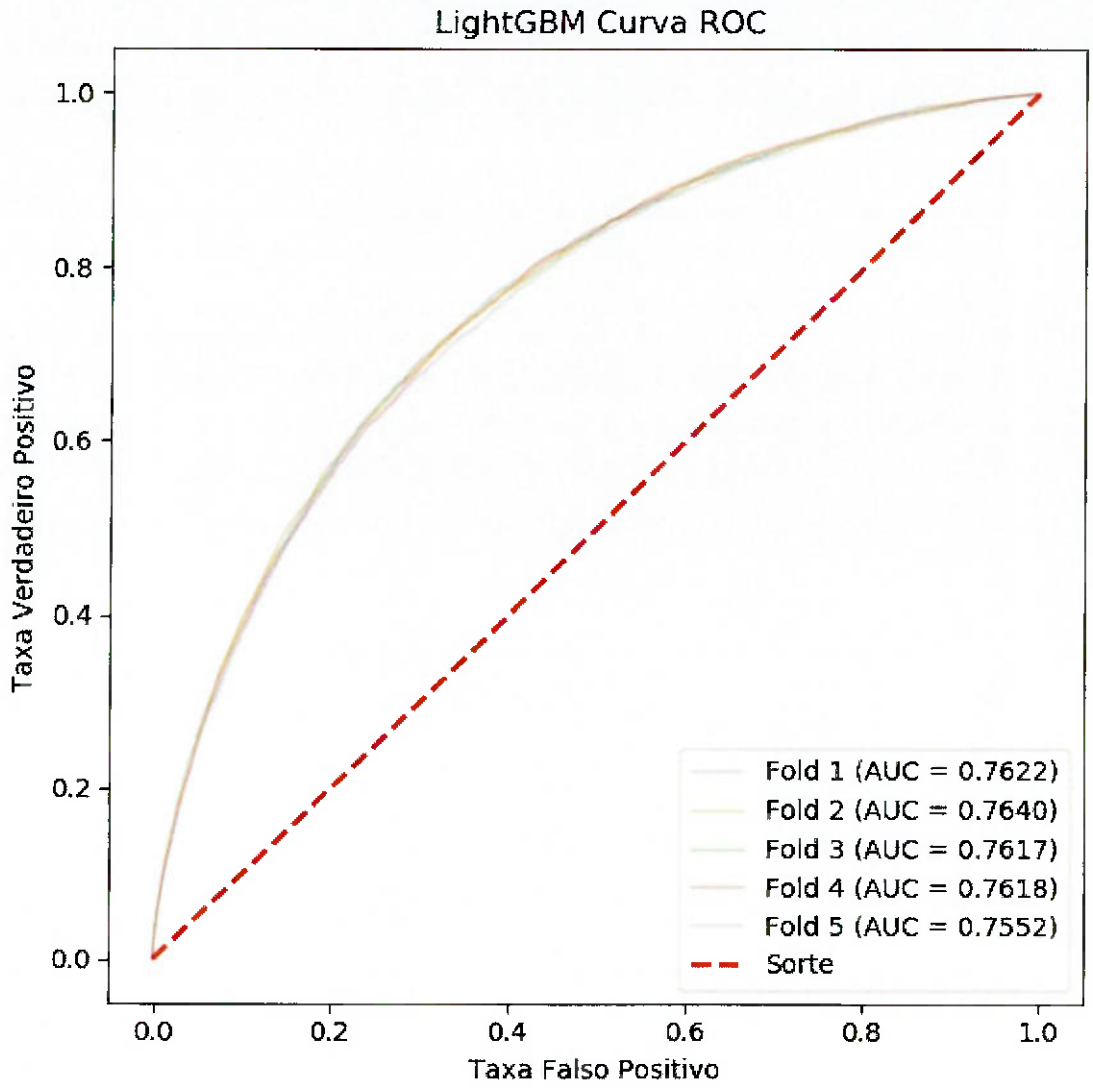
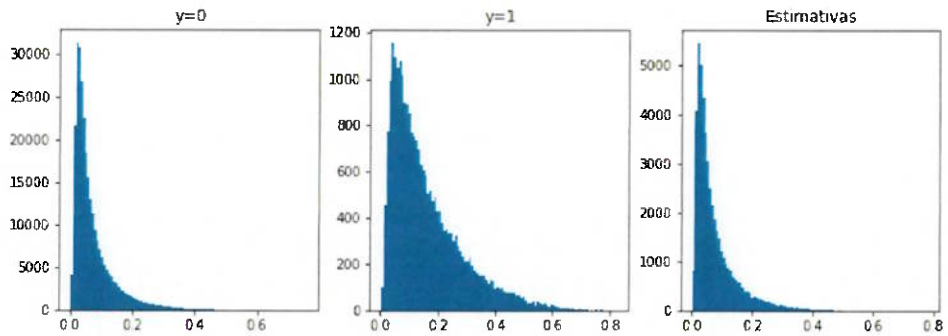


Figura 16 – Distribuiçao de Probabilidade para cada Classe



Na figura 14 o modelo executa o procedimento de ajuste aos dados até o momento que a métrica de performance (*score*) não é maximizada nas ultimas 100 (cem) iterações

(*rounds*). Este procedimento de parar o treinamento representa o uso do método de *early stop*. A métrica é resultado do ajuste aos dados de treinamento, e aos dados de validação cruzada com uso do método de *stratified k-fold* e uso de 5 (cinco) *folds*. A pontuação calcula a métrica *ROC AUC* estabelecida na competição e a função objetivo *proxy* utilizada foi a *logloss*. Existe um problema de otimização explicado no capítulo 2 com a métrica *ROC AUC* onde não é possível obter a derivada da métrica e outra função deve ser utilizada. Não foi obtido nenhuma vantagem no uso de outras funções diferentes da *logloss*.

Todos modelos foram treinados com uso de *early stop*, até que a métrica da média da validação cruzada não fosse maximizada nos últimos 100 (cem) ajustes executados. No ajuste final é executado um procedimento parecido, com o número de iterações escolhido quando o valor da métrica de validação cruzada de todos os *folds* não é incrementado. Este procedimento foi executado pois todo conjunto de dados foi utilizado, e uma otimização em um número de iterações maior pode ser utilizado, com risco de um possível *overfitting*. A diferença de iterações não foi superior a 50 (cinquenta) em relação ao uso da média das métricas, e os valores de validação estavam próximos. Foi levantado as métricas de performance para os modelos e descartados as métricas que não tiveram performance próximas aos modelos finais e métricas inferiores a *baseline*.

Para avaliação visual da performance foi utilizada a figura 15 da curva *ROC*. No primeiro modelo utilizado para métrica *baseline*, a métrica *ROC AUC* ficou próxima de 0,760 com um *fold* contendo métrica inferior a média. Todos modelos que tiveram valor superior a métrica *baseline* foram considerados na seleção de modelos. A composição da curva *ROC* foi criada com base na distribuição de probabilidade do modelo para as classes 0 e 1 conforme figura 16, onde a primeira imagem se refere as probabilidades calculadas para a classe 0, a segunda imagem para classe 1, e a terceira a distribuição de todas probabilidades calculadas.

No caso prático, o procedimento foi executado inúmeras vezes, a cada novo ciclo uma variável foi alterada, excluída ou incluída. Para a inclusão e criação de variáveis foi utilizado a relevância das variáveis do modelo, onde a importância das variáveis (*features*) do modelo é ranqueada. Esta métrica pode ser apresentada como o resultado da variável g da função objetivo de otimização do modelo conforme explicado no capítulo 2, ou o número de vezes que o modelo de árvore de decisão utilizou a variável para criar um novo nó. Foi utilizado a métrica do número de utilizações da variável pelo modelo.

Um conjunto de variáveis relativas a pontuações de "birôs" de crédito tiveram uma importância relevante. A importância excessiva dessas variáveis pode eliminar o uso das demais, e o modelo se torna mais sensível e dependente delas. Caso um "birô" de maneira tendenciosa produza um resultado falso, o modelo pode utilizar esta informação de forma enviesada. Este conhecimento do fenômeno físico e da sensibilidade do modelo é importante, pois na "vida real" é necessário atenção para possíveis *outliers* das variáveis mais sensíveis, evitando uma possível manipulação do modelo.

Foram realizadas mais de 300 (trezentas) iterações com métrica superior ao *baseline*. Foi utilizado a otimização *bayesiana* que é livre do uso de derivadas para buscar melhores parâmetros do modelo ajustado. Não é o objetivo explicar o funcionamento da otimização *bayesiana*. Foi utilizado também a iteração manual para compreender os parâmetros do modelo que tinham mais sensibilidade aos dados. Este processo melhorou a performance dos modelos sem necessidade de criar novas variáveis.

A seguir um exemplo de um conjunto de resultados de uma otimização *bayesiana* no conjunto de parâmetros do modelo:

Tabela 2 – Otimização *Bayesiana* dos Parâmetros

Passo	Tempo	AUC	colsample	max_depth	min_child_weight	subsample
1	06m46s	0.78378	0.2762	7.9750	31.3170	0.7055
2	09m25s	**0.78524**	0.4175	2.2879	27.2413	0.7351
3	03m48s	0.78371	0.3009	5.4172	20.4099	0.5636
4	03m41s	0.78299	0.6522	8.4458	43.6216	0.9912
5	07m53s	0.78446	0.6224	2.7637	3.28820	0.5102
6	04m04s	0.78378	0.2762	7.9750	31.3170	0.7055
7	09m17s	**0.78524**	0.4175	2.2879	27.2413	0.7351
8	03m28s	0.78251	0.2762	7.9750	31.3170	0.7055
9	14m02s	0.78481	0.4175	2.2879	27.2413	0.7351

As linhas representam um resultado de uma validação cruzada informada pela coluna "Passo". A métrica da validação cruzada é representada pela coluna "AUC", as demais colunas são referentes aos parâmetros do modelo, e o tempo gasto para executar a validação cruzada. Pode-se verificar que a melhor métrica de validação cruzada neste exemplo foi de 0,78524 nas linhas 2 (dois) e 7 (sete). Observa-se que alguns parâmetros são apresentados como números contínuos, porém o parâmetro do modelo utilizado é um número inteiro. O número foi truncado para representar um número inteiro na execução do modelo, porém foi utilizado um valor contínuo para otimização.

Foi treinado na etapa 6 (seis) os modelos sobre todos os dados utilizando os parâmetros da otimização *bayesiana*, com observação quanto ao método de *early stop* explicado a cima. Uma estimativa foi realizada para cada modelo e enviada para plataforma de competição. A título de curiosidade foi enviado um ultimo modelo com a média aritmética de todos os 6 (seis) modelos enviados. O resultado dos 6 (seis) melhores modelos serão avaliados no próximo capítulo.

5 Apresentação e discussão dos resultados

5.1 Apresentação dos resultados

As métricas dos modelos que tiveram performance inferior a métrica *baseline* foram descartadas. Também foram descartadas os modelos que tiveram métricas muito menores que os 6 (seis) melhores modelos apresentados.

Para avaliação dos 6 (seis) melhores modelos, será utilizada a representação em tabela:

Tabela 3 – Resultado dos modelos enviados

Modelo	Treinamento	**Validação Cruzada**	**Placar Privado**	**Placar Público**
Média	-	-	0.78968	0.79366
LGB	0.8510	0.7988	**0.79175**	**0.79596**
LGB	0.8510	0.7968	0.79163	0.79572
LGB	0.8810	**0.7999**	0.79170	0.79588
XGB	0.8880	0.7934	0.78693	0.78550
XGB	**0.9031**	0.7955	0.78676	0.78582
XGB	0.8706	0.7960	0.78655	0.78532
Baseline	-	0.760974	-	-

A tabela mostra em ordem temporal as estimativas utilizadas para avaliar os modelos finais. Os modelos utilizados são apresentados na primeira coluna: "Média" representando a média aritmética das estimativas dos 6 (seis) modelos, "XGB" representando o modelo *XGBoost*, "LGB" representando o modelo *LightGBM* e modelo "*Baseline*" é apresentado e não foi escolhido como melhor modelo. Os resultados das métricas de performance dos modelos são apresentados nas demais colunas: a segunda coluna é o ajuste do modelo aos dados de treinamento, a terceira coluna a validação cruzada, a quarta coluna as estimativas enviadas para competição no placar "Privado" e na quinta coluna as estimativas para competição no placar "Público".

O modelo "Média" foi enviado a título de curiosidade para avaliar o efeito da média das estimativas dos modelos. Nota-se que a média não foi o melhor modelo e este não foi avaliado pelas etapas de validação cruzada.

5.2 Análise crítica dos resultados obtidos e conclusões

Os resultados foram satisfatórios para o objetivo do trabalho.

Pode-se observar que os valores de validação cruzada estão próximos aos valores da competição. Esta avaliação remete ao exposto em (KOHAVI, 1995). Os modelos apresentam uma pequena diferença entre a validação cruzada e o placar "Privado", em torno de 0,01 na métrica *ROC AUC*, uma diferença próxima a 1,25%. Os modelos finais apresentam uma diferença em torno de 0,03 na métrica *ROC AUC*, uma diferença próxima de 3,8%.

Pode-se notar que os valores das métricas foram superiores ao caso especial da *ROC AUC* de valor próximo à 0,5. Isto demonstra que o modelo tem capacidade discriminativa das classes. Era esperado não obter um modelo ideal de métrica 1.

É observado que o valor da métrica no conjunto de treinamento é alta. A diferença é de certa forma preocupante pois o modelo precisou executar um ajuste muito próximo aos dados de treinamento, sendo possível um *overfitting*. Porém vale ressaltar que os seis melhores modelos obtidos passaram por otimização *bayesiana* e ajuste manual dos parâmetros. Foi comparado os modelos com parâmetros próximos aos obtidos na otimização, e eles não resultaram em variações bruscas nas métricas de validação cruzada. Esta foi uma forma utilizada para avaliar se os modelos estariam sendo influenciados por *overfitting*, e felizmente a métrica de validação cruzada se manteve estável.

Estes modelos poderiam ser utilizados em ambiente real, devendo ser avaliado a questão da perda esperada dada pelos demais fatores LGD e EAD.

Paralelamente ao resultado do trabalho, o resultado da competição foi muito satisfatório, dado o tratamento não intensivo sobre a criação de variáveis. Foi obtido uma pontuação entre os 30% (trinta porcento) dos melhores modelos da competição. Com uma pequena diferença dos competidores próximos em número equivalente a 0,00001 da métrica *AUC* de 0,79175, ou seja, 0,001%. O primeiro colocado teve um resultado de 0,80570, uma diferença de 0,01395 ou 1,762% na métrica *AUC*. A metodologia utilizada pelo primeiro colocado está disponível no portal *Kaggle* e foi realizado por 5 (cinco) participantes, o segundo colocado com uma métrica de 0,80561 possui um total de 12 (doze) participantes entre eles os melhores competidores do portal. O terceiro competidor relata que possui experiência profissional na área de crédito, e obteve uma métrica de 0,80511 com uma equipe de 2 (dois) participantes.

6 Conclusões

6.1 Resultados gerais

Pode-se afirmar que a metodologia utilizada e o uso de validação cruzada exposto em (KOHAVI, 1995) apresentaram resultados satisfatórios com os dados utilizados. O resultado geral da competição também foi satisfatório, considerando a escolha de não "estressar os dados", a falta de especialização na área e experiência sobre avaliação de crédito imobiliário.

O processo de modelagem seguiu as etapas definidas no capítulo 3.3, fundamentadas pelas referências bibliográficas expostas no capítulo 2. O trabalho demandou vários dias de modelagem até a obtenção dos modelos finais. Não foram tratados os arquivos com séries de dados temporais onde foram utilizados somente a primeira ocorrência de cada arquivo. Esta informação descartada pode gerar melhor representação das características dos clientes que incorreram em *default*.

A utilização de validação cruzada foi de extrema importância em todas etapas. A utilização da informação de relevância de variáveis fornecida pelos modelos de árvore de decisão, foi de extrema importância para escolha das interações de variáveis. O ajuste dos parâmetros dos modelos pela otimização *bayesiana*, auxiliou a explorar um conjunto de valores dos parâmetros que não foram utilizados inicialmente. Todos estes fatores apresentaram uma melhora importante no resultado dos modelos. A utilização de ajuste manual auxiliou a verificar a presença de *overfitting* nos dados de treinamento, com a verificação da métrica de validação cruzada se mantendo estável.

Ressalta-se que não foi utilizado informação do placar "Público" para melhorar a métrica de validação cruzada e obtenção de melhores parâmetros dos modelos. O processo de criação de variáveis não foi exaustivo ou criativo, foram utilizados divisão de variáveis, média de variáveis e contagem de campos nulos. A remoção de variáveis que apresentaram pouca relevância e não apresentaram alteração significante foi verificada pela métrica de validação cruzada.

Algumas iterações levaram a modelos abaixo da métrica de *baseline*, métrica do modelo no conjunto mínimo e inicial de variáveis e parâmetros. Em alguns casos a métrica baixa não teve uma explicação conclusiva e o modelo foi descartado. O número de modelos ajustados foi superior a 300 (trezentos), este número não é grande pois muitos ajustes

foram devidos a otimização *bayesiana* que ocorreu sem a intervenção humana. A etapa que demandou maior tempo foi da criação de variáveis e é conhecido a sua importância no resultado final.

6.2 Conclusão sobre as variáveis

É importante relatar que as conclusões sobre os dados estão muito em linha com oque foi observado em (LAWRENCE L. DOUGLAS SMITH, 1992). As variáveis utilizadas foram obtidas conforme procedimento explicado no capítulo 4, e foi observado que a variável de maior importância para o modelo é a relação do empréstimo dividido pelo valor do bem. A relação de cobertura do pagamento não foi avaliada de maneira profunda pois foi utilizado somente a primeira linha do arquivo. O padrão de dados nulos e a média de algumas variáveis apresentou alguma melhoria, mas os resultados podem ser apenas marginais perto da relevância das demais relações.

É importante relatar que o uso de bibliografia específica na área de conhecimento auxilia na etapa de compreensão das variáveis e na melhora final do modelo. Trabalhos na área de crédito são poucos discutidos e as variáveis relevantes são dificeis de serem encontradas e explicadas. O material de (LAWRENCE L. DOUGLAS SMITH, 1992) tem uma relevância importante no trabalho pois apresentou uma conclusão em linha com o obtivo pelo processo "mecânico" de criação das variáveis e avaliação dos modelos.

7 Pesquisas Futuras

Conforme apresentado na introdução do trabalho, não foi executado a modelagem da perda esperada e seus fatores: EAD e LGD. Este é um próximo passo para gerenciamento de risco de crédito das instituições. A possibilidade de expansão do trabalho é enorme. Pode-se estimar a renda necessária para um empréstimo, melhores calendários para pagamento das parcelas, melhor precificação das taxas de juros utilizadas, avaliar os casos em que foram negado empréstimos e fomentar a venda do bem para outros clientes que seriam atendidos, entre várias outras possibilidades que também são mencionadas em (LAWRENCE L. DOUGLAS SMITH, 1992).

Para pessoas que estudam aprendizado de máquina, a plataforma é um ambiente excelente para aprendizado. No portal *Kaggle* é possível verificar as abordagens relatadas pelos demais competidores. Existem equipes que relatam mais de 20 (vinte) modelos agregados, uso de redes neurais, redução de dimensionalidade com *T-SNE*[1], *PCA*, e outras técnicas.

Com objetivo de pesquisa acadêmica ou industrial, é interessante avaliar quais destas publicações expostas pelos melhores participantes são possíveis de serem utilizadas na "vida real". A busca tem por objetivo encontrar modelos simples com boa performance e metodologia aplicada, ou explicações sólidas sobre as variáveis envolvidas e conhecimento sobre o fenômeno físico. Também é interessante avaliar como foram utilizados os conjuntos de dados descartados neste trabalho, que se referem a séries temporais. Com foco nestes itens, é recomendável a avaliação da terceira melhor equipe representada por dois competidores, onde um trabalhou para instituição bancária com conhecimento sólido de problemas de avaliação de crédito.

[1] *t-Stochastic Neighbor Embedding*

Referências[2]

ADDO DOMINIQUE GUEGAN, B. H. P. M. *Credit Risk Analysis Using Machine and Deep Learning Models.* 2018. Disponível em: <https://www.mdpi.com/2227-9091/6/2/38/pdf>. Citado 2 vezes nas páginas 37 e 38.

CREDIT, H. *Kaggle Competition.* 2018. Disponível em: <https://www.kaggle.com/c/home-credit-default-risk>. Citado na página 49.

ELSEVIER. *Journal of Banking and Finance.* 2018. Disponível em: <https://www.journals.elsevier.com/journal-of-banking-and-finance>. Citado na página 35.

GOOGLE. *Google Acadêmico.* 2018. Disponível em: <https://scholar.google.com.br/>. Citado na página 35.

HASTIE ROBERT TIBSHIRANI, J. F. T. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Second Edition.* [s.n.], 2017. Disponível em: <https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf>. Citado na página 21.

KOHAVI, R. *A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection.* 1995. 1137-1145 p. Disponível em: <http://web.cs.iastate.edu/~jtian/cs573/Papers/Kohavi-IJCAI-95.pdf>. Citado 7 vezes nas páginas 14, 20, 23, 43, 46, 58 e 59.

LAWRENCE L. DOUGLAS SMITH, M. R. E. C. *An analysis of default risk in mobile home credit.* 1992. Disponível em: <https://doi.org/10.1016/0378-4266(92)90016-S>. Citado 5 vezes nas páginas 36, 37, 38, 60 e 61.

MICROSOFT. *LightGBM - Features.* 2018. Disponível em: <https://lightgbm.readthedocs.io/en/latest/Features.html>. Citado 3 vezes nas páginas 14, 33 e 35.

NETS comp.ai.neural. *Section - What are cross-validation and bootstrapping?* 2018. Disponível em: <http://www.faqs.org/faqs/ai-faq/neural-nets/part3/section-12.html>. Citado na página 21.

PRATI, G. E. A. P. A. B. e. M. C. M. R. C. *Curvas ROC para avaliação de classificadores.* 2008. Disponível em: <http://conteudo.icmc.usp.br/pessoas/gbatista/files/ieee_la2008.pdf>. Citado 2 vezes nas páginas 14 e 15.

RAJNARAYAN, D. W. D. *Bias-Variance Trade-offs: Novel Applications.* 2018. Disponível em: <https://arxiv.org/pdf/0810.0879v1.pdf>. Citado na página 23.

RESEARCH, K. *Receiver Operating Characteristic (ROC) Curves.* 2016. Disponível em: <https://kennis-research.shinyapps.io/ROC-Curves/>. Citado na página 16.

SSRN. *Social Science Research Network.* 2018. Disponível em: <https://papers.ssrn.com/>. Citado na página 35.

WIKIPEDIA. *Validação Cruzada.* 2018. Disponível em: <https://pt.wikipedia.org/wiki/Valida%C3%A7%C3%A3o_cruzada>. Citado na página 22.

[2] De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

XGBOOST. *Introduction to Boosted Trees.* 2016. Disponível em: <https://xgboost.readthedocs.io/en/latest/tutorials/model.html>. Citado 5 vezes nas páginas 14, 27, 28, 29 e 33.