Universidade Federal do Espírito Santo
Programa Institucional de Iniciação Científica
Relatório Final de Pesquisa
Ciências Exatas e da Terra

Interpretabilidade em modelos preditivos na área da saúde

Identificação:

Grande área do CNPq: Ciências Exatas e da Terra
Área do CNPq: Probabilidade e Estatística
Título do Projeto: Ciência de Dados e Aprendizado Estatístico Aplicados à Saúde
Professor Orientador: Profª. Drª. Agatha Sacramento Rodrigues
Estudante PIBIC/PIVIC: Ornella Scardua Ferreira

Resumo: *Na abordagem preditivista no ajuste de modelos, o interesse consiste na construção de uma regra para predizer novas observações. Em uma classe de modelos candidatos, é escolhido aquele que apresenta o melhor desempenho preditivo e acurácia (capacidade em acertar uma predição ou errar dentro de um limiar aceitável). O modelo vencedor pode ser um modelo não explicável, ou seja, um modelo cujas decisões não podem ser explicadas (modelo caixa preta). No entanto, destaca-se a importância em entender o motivo e os fatores que levam a certas previsões ou comportamentos, além de ser mais fácil para os humanos confiarem em um sistema que explique suas decisões. Nesse cenário, métodos de interpretabilidade se fazem necessários, uma vez que podem ser aplicados a qualquer modelo preditivo previamente ajustado. Com esse estudo se objetiva estudar métodos de interpretabilidade globais e individuais aplicados nas decisões de qualquer modelo preditivo, comparando-os em aplicações da área da saúde e discutindo suas vantagens e desvantagens em diferentes cenários postos. Esse projeto também tem como objetivo disponibilizar ferramentas para análise dos métodos, por meio de documentação computacional bem formulada, disponível no GitHub e por uma plataforma web (aplicativo Shiny). Os métodos de interpretabilidade serão aplicados a um conjunto de dados real do Departamento de Obstetrícia da Faculdade de Medicina da Universidade de São Paulo (FM-USP). Com esse projeto, espera-se discutir a utilização de modelos preditivos na área da saúde disponibilizando para os usuários métodos de interpretabilidade das decisões do modelo previamente ajustado, e dessa forma permitir que o melhor modelo para um problema em questão seja utilizado (ainda que este seja um modelo caixa preta).*

Palavras chaves: Aplicações na área da saúde; Aplicativo Shiny; Métodos de explicabilidade individuais; Métodos de explicabilidade globais; Modelos preditivos.

1 Introdução

Com o aumento da capacidade de armazenamento e processamento de dados, sua exploração e análise exigem não apenas métodos estatísticos, mas também de técnicas computacionais. A área de *Machine Learning* (em português Aprendizado de Máquina ou Automático) é um resultado dessa interação entre a Estatística e a Computação, utilizando modelos estatísticos combinados com algoritmos computacionais para extrair informação de conjuntos de dados com muitas observações e/ou variáveis. ML pode ser supervisionado ou não supervisionado (Morettin, 2020), e consideramos aqui ML supervisionado, o qual engloba modelos para estudar o valor de uma variável resposta (*output*, *label* ou desfecho) a partir de cováriaveis (*input*, *features*, variáveis explicativas ou preditoras).

Em ML supervisionado, Breiman (2001) argumenta que uma distinção entre duas culturas precisa ser feita. A primeira cultura, chamada de modelo explicativo (alguns autores chamam de modelo inferencial), foca na interpretação dos parâmetros envolvidos do modelo e testa hipóteses para entender a relação entre as covariáveis e a variável resposta. Sob esta abordagem, testar se as suposições do modelo (por exemplo, normalidade dos erros, linearidade, homoscedasticidade etc) são válidas é de fundamental importância, uma vez que o objetivo, em geral, está na inferência. Na segunda cultura, chamada de *algorithmic modeling culture* por Breiman, o principal objetivo é a construção de um modelo (regra) para predizer novas observações (Izbicki, 2020).

Historicamente, modelos explicativos são amplamente utilizados na área da saúde para entender, por exemplo, que a exposição a um dado fator tem tantas vezes mais chance de ter o desfecho de interesse e que essa relação é importante do ponto de vista estatístico e clínico. Nos últimos anos, tem surgido o interesse em também realizar predições de desfechos na área da saúde e modelos preditivos têm sido cada vez mais utilizados. Os exemplos podem ser listados: detecção de câncer de cólon derivada de colonoscopias virtuais em 3D, predição da sobrevida hospitalar em curto prazo em pacientes com lúpus e encontro das características clínicas ou demográficas mais preditivas para pacientes com artrite espinhal (Malley et al., 2011).

A importância em definir o objetivo do modelo antes de seu ajuste se deve ao fato de que o processo de modelagem é diferente a depender da intenção. Quando o intuito é inferencial, as escolhas feitas durante o processo de modelagem são pautadas em medidas que avaliam a relação e a força da explicação entre as variáveis. Já com o objetivo de predição, as escolhas no processo de modelagem são guiadas por medidas de desempenho preditivo e acurácia do modelo (capacidade em acertar uma predição ou errar dentro de um limiar aceitável). Nesse último caso, o modelo escolhido é aquele com melhor desempenho preditivo, e esse “melhor” pode ser um modelo não explicável, ou seja, um modelo cujas decisões não podem ser explicadas, uma vez que seu funcionamento interno não pode ser facilmente acessado.

No entanto, mesmo com o intuito de predição, muitos pesquisadores e médicos têm o interesse em entender as variáveis no modelo e discutir o porquê delas e suas relações nesse modelo preditivo. Para facilitar, então, o aprendizado sobre o motivo e fatores mais importantes

de certas previsões ou comportamentos, a interpretabilidade e as explicações das decisões são cruciais (Molnar, 2019). Nesse sentido, foram propostos métodos de interpretabilidade, os quais podem ser aplicados a qualquer modelo preditivo previamente ajustado.

Um desses métodos é o SHAP (*SHapley Additive exPlanations*), que estuda o impacto das covariáveis na saída do modelo, usando todas as combinações possíveis de presença e ausência das covariáveis, e explica individualmente as decisões do modelo (Molnar, 2019). O novo escore de crédito da Serasa (pontuação de pessoa física que quantifica a sua propensão de ser uma boa pagadora), por exemplo, apresenta um campo que explica os fatores que aumentam e diminuem a pontução obtida pela pessoa física (PF), algo inexistente no escore de crédito anterior da empresa. Essa explicação individual (para cada PF) é obtida por algum método de explicabilidade, possivelmente o SHAP (https://www.serasa.com.br/score/). Outros métodos de explicabilidade também serão considerados, comparados e avaliados nesse projeto, e serão melhor descritos na seção ”Metodologia”.

Se as decisões de um modelo de ML podem ser explicadas, as seguintes questões podem ser checadas mais facilmente (Doshi-Velez & Kim, 2017): 1) justiça (*fairness*) – garantir que as previsões sejam imparciais, não viesadas e que não discriminem grupos sub-representados; 2) confiabilidade - testar se pequenas alterações na entrada não levam a grandes alterações na previsão; 3) causalidade – verificar se apenas os relacionamentos causais são detectados; 4) confiança - é mais fácil para os humanos confiarem em um sistema que explica suas decisões.

Nesse projeto, modelos preditivos na área da saúde e métodos de interpretabilidade em uma aplicação da área da medicina obstétrica serão discutidos. Nesta aplicação, deseja-se predizer, no momento do diagnóstico de diabetes gestacional, se uma gestante fará o uso de insulina em algum momento antes do parto com base em informações clínicas, exames laboratoriais, histórico obstétrico e familiar. Aplicação, esta, resultante de pesquisas realizadas no Departamento de Obstetrícia da Faculdade de Medicina da Universidade de São Paulo (FM-USP), em particular pelos ambulatórios de diabetes gestacional.

Esse subprojeto é ligado ao projeto de pesquisa “Ciência de Dados e Aprendizado Estatístico Aplicados à Saúde” (número de registro 10225/2020 e certificado no CNPq), coordenado pela professora orientadora, e que se objetiva em aplicar e desenvolver metodologias na área de Ciência de Dados para resolver problemas e inovação na área da saúde.

No Capítulo 3 é descrita a metodologia, na qual são postas as discussões acerca das abordagens estatísticas e de *machine learning* propostas para alcançar os objetivos desse projeto. No Capítulo 4 estão as aplicações em *software* `R` e os resultados obtidos em cima de uma base de dados composta por 31 variáveis e 408 observações, cujos indivíduos são gestantes diagnosticadas com diabetes gestacional que foram consultadas pelos ambulatórios de diabetes gestacional do Hospital das Clínicas da FM-USP. Por fim, no Capítulo 5 encontram-se as discussões finais e conclusão com relação aos efeitos das métricas de interpretabilidade sobre o modelo preditivo cuja acurácia foi maior.

2 Objetivos

O objetivo geral desse projeto é avaliar alguns métodos de interpretabilidade em diferentes modelos preditivos em uma aplicação real da área da saúde.

São os objetivos específicos:

1. Avaliar os modelos preditivos explicáveis e não explicáveis, discutindo suas aplicações em problemas da área da saúde.

2. Comparar os métodos de interpretabilidade, identificando suas vantagens e desvantagens sobre um modelo preditivo previamente ajustado.

3. Discutir a *fairness*, confiabilidade, causalidade e confiança a partir das decisões de modelos preditivos.

4. Documentar os pacotes e códigos computacionais de métodos de interpretabilidade para a aplicabilidade dos métodos estudados.

5. Desenvolver uma plataforma web por meio de um aplicativo Shiny (https://www.shinyapps.io/) para melhor acessibilidade dos métodos de interpretabilidade para os usuários.

6. Apresentar e discutir os resultados para os grupos de pesquisa do Departamento de Obstetrícia da FM-USP.

3 Metodologia

Em um cenário de ML supervisionado com foco em predição, problemas de sobreajuste (*overfitting*) são bastante comuns. Normalmente, o sobreajuste ocorre em modelos considerados muito complexos cujos ajustes se adequam perfeitamente aos dados que serviram para definir esses modelos, mas que são incapazes de generalizarem seus resultados sobre novas observações.

Para identificar se um modelo está com problema de sobreajuste, deve-se avaliar a performance preditiva em cima de dados novos: se a performance cair consideravelmente, há fortes indícios de que o modelo esteja sobreajustado. No entanto, obter novos dados na prática é uma tarefa quase sempre inviável. Nessa situação, uma solução é aplicar o método de validação cruzada, um conjunto de técnicas que calcula o erro de teste de modelos estatísticos a partir da divisão da amostra original em amostra de treino e amostra de teste (James & Tibshirani, 2013). Neste estudo, consideramos a validação cruzada do tipo *k-fold* (em inglês *k-fold cross validation*) - outros tipos podem ser vistos em Amorim (2019).

Com a amostra de treinamento se faz o ajuste do modelo, enquanto na amostra de teste (ou de validação), que funciona como um subconjunto de observações novas, é calculado o erro de teste (a depender de seu valor, verifica-se se o modelo teve bom desempenho ou não). Usualmente, os dados para treinamento correspondem a cerca de 70 a 80% e os dados que serão testados em 20 a 30% da amostra original. Uma vez divididos os dados aleatoriamente em treino e teste, pegam-se os dados usados para treinar e subdivide-os, também de forma

aleatória, em k amostras disjuntas e aproximadamente com o mesmo tamanho. Em cada k ajuste do modelo é selecionado k-1 subconjuntos para treino e um único subconjunto para teste, o qual é diferente em cada k iteração. Finalizada a rodagem do subconjunto de teste, a performance preditiva do modelo é obtida por meio da média dos erros de teste que foram calculados em cada ajuste. Formalmente, é definida por

$$P_{k-fold} = \frac{1}{k}\sum_{t=1}^{k} EQM_i = \frac{1}{k}\sum_{i=1}^{k} \frac{(y_i - \hat{y}_i)^2}{n_t},$$

em que y_i e $\hat{y}_i$ é, respectivamente, o valor observado e o valor predito da i-ésima observação e n_t é o número de observações no t-ésimo subconjunto de teste.

A notação utilizada em modelos de ML supervisionados e preditivistas é descrita no que segue. Seja uma amostra observada de n indivíduos e sejam (x_i, y_i) os dados para o i-ésimo indivíduo, com $i = 1,...,n$, em que $x_i = (x_{1i}, x_{2i}, ..., x_{pi})^\top$ é o vetor de p covariáveis e y_i denota a observação da variável resposta.

O conjunto de modelos possíveis de ajuste depende das características dos dados, principalmente da variável resposta. Quando a variável resposta Y é uma variável quantitativa, o problema é de regressão, e quando Y é qualitativa, o problema é de classificação.

Dentre os modelos de regressão, existem os modelos paramétricos (modelo linear, Gama e Poisson são alguns exemplos) com e sem regularização (por exemplo, *Ridge* e Lasso) e os métodos não-paramétricos, como Árvores de Regressão, *Bagging* e Florestas Aleatórias, *Boosting* e *Support Vector Regression*. Para classificação, alguns modelos podem ser citados, como os modelos de Regressão Logística, Regressão Multinomial, Análise Discriminante, *Naive Bayes*, *Support Vector Machines* (SVM), Árvores de Classificação, método dos K-Vizinhos mais Próximos e Redes Neurais (Izbicki, 2020).

Muitos problemas nos quais modelos de ML estão envolvidos têm como objeto de estudo uma pontuação medida sob uma escala numérica contínua ou uma probabilidade associada a determinado evento de interesse, em que a partir de certo valor ou probabilidade é possível discriminar a variável resposta em duas ou mais classes. São alguns exemplos: na medicina é comum atribuir valores a exames, e conforme esse valor, classificar como positivo ou negativo para uma doença (Margotto, 2010); no sistema de saúde dos EUA, uma pontuação de risco é fornecida a um paciente segundo seu histórico de saúde, e com base nessa pontuação, o paciente será inscrito em um programa de saúde extensivo imediatamente ou dependerá de uma nova avaliação médica para ser contemplado pelo programa (Obermeyer et al., 2019); a Serasa classifica o consumidor como bom ou má pagador de acordo com o escore de crédito, que varia de 0 a 1000 (https://www.serasa.com.br/ensina/aumentar-score/score-de-credito/).

Em comum entre esses exemplos existe a definição de um limiar que determina se o elemento vai apresentar a característica de interesse ou não. Esse limiar, também chamado de *cutoff point*, *threshold* ou ponto de corte, é um parâmetro da Curva ROC (*Receiver Operating Characteristic*) que estabelece a relação entre as medidas de sensibilidade e especificidade.

Considere um problema de classificação em que Y é binária e assume valores em $\{0,1\}$, com $y=0$ e $y=1$ denotando, nesta ordem, a ausência e a presença da característica de interesse. A sensibilidade, também conhecida como a taxa dos verdadeiros positivos (TVP), é a probabilidade de decisões $\hat{y}$ tomadas corretamente dentre as observações que têm o evento de interesse, isto é,

$$TVP = p(\hat{y}=1|y=1) = \frac{VP}{VP+FN},$$

em que VP são os verdadeiros positivos e FN são os falsos negativos. Já a especificidade, ou a taxa dos falsos positivos (TFP), se caracteriza pela probabilidade de decisões corretas entre aquelas observações que não apresentaram o evento de interesse, ou seja,

$$TFP = p(\hat{y}=0|y=1) = \frac{FP}{FP+VN},$$

com FP sendo os falsos positivos e VN os verdadeiros negativos.

Certamente, o modelo não acertará 100% em suas decisões. Por essa razão, deseja-se um limiar α que melhor minimiza os erros de classificação tão quanto for possível. A Curva ROC surge como uma ferramenta gráfica simples mas poderosa para tal, sendo possível obter a capacidade de predição para diferentes valores de α ao mesmo tempo em que se torna evidente o melhor limiar que otimiza a sensibilidade em função da especificidade. O ponto ótimo da curva se encontra sempre mais próximo do canto superior esquerdo do gráfico, pois isso significa que TVP está mais próxima de 1 e TFP está mais próxima de zero.

Com a combinação de limiares, sensibilidade e especificidade, tem-se a Área Sob a Curva ROC (em inglês *Area Under the ROC Curve* - AUC), que nada mais é que a simplificação desses resultados condensados em um valor de acurácia entre zero e 1. A área abaixo da Curva ROC corresponde a um quadrado de lado 1, e ela estar mais próxima de 1 se traduz em melhor desempenho preditivo para o modelo (Figura 1).

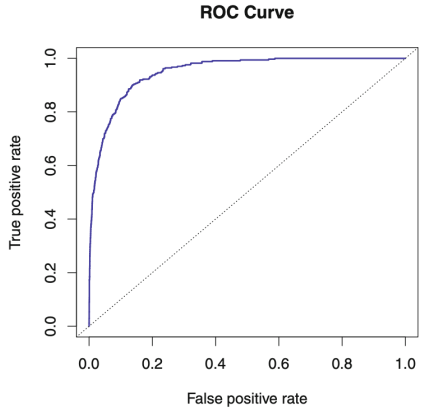

Figura 1: Gráfico da Curva ROC. Fonte: James et al. (2013).

Apesar da popularidade dos modelos de *machine learning*, muitos deles permanecem sendo não-interpretáveis (modelos caixa-preta ou *black-box models*, em inglês). Em outras palavras,

são sistemas fechados e complexos que inviabilizam o entendimento da função que é modelada pelo algoritmo desses modelos.

Na prática, isso significa que não conseguimos entender a relação entre as covariáveis e a variável resposta de forma direta, como acontece em um modelo de regressão logística, por exemplo. No entanto, compreender os motivos por que ocorreram certas previsões é adequado para se fazer inferência (Breiman, 2001) e muito importante para avaliar a justiça (*fairness*) e a confiança de um modelo (Doshi-Velez & Kim, 2017).

Em razão disso, foram propostos métodos de interpretabilidade em que as interpretações das decisões de um modelo caixa-preta podem ser individuais, isto é, para cada observação, ou globais, quando a interpretação é feita em termos de média.

Vamos considerar e descrever de forma resumida alguns modelos preditivos de classificação na seção 3.1 e alguns modelos de interpretabilidade global e individual na seção 3.2.

3.1 Modelos de predição

3.1.1 Regressão Logística

Neste método estatístico em específico, a variável resposta é dicotômica e não contínua, ou seja, Y é binário assumindo valores $\{0,1\}$. Geralmente, $y_i = 1$ é atribuído a uma resposta positiva, que quer dizer a presença do evento de interesse, e $y_i = 0$ se refere a uma resposta negativa, que significa a ausência do evento.

Nesse contexto, Y_i dado um vetor coluna de observações x_i tem distribuição Bernoulli com parâmetro p desconhecido, isto é,

$$Y_i|X_i = x_i \sim Ber(p).$$

De modo geral, o objetivo da Regressão Logística está em estimar a probabilidade desconhecida p em função de uma combinação linear de covariáveis x_i independentes. Hosmer e Lemeshow (2000) afirmam que ao considerar como função logito a função de ligação de p ao preditor linear $(\beta_0 + \beta_1 + \ldots + \beta_p)$, este modelo pode ser expresso como a probabilidade de sucesso

$$p(x_i) = P(Y_i = 1|X = x_i) = \frac{exp(\beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{ip})}{1 + exp(\beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{ip})}, \quad i = 1, \cdots, n, \tag{1}$$

em que $x_i = (x_{1i}, \ldots, x_{ip})$.

Assim, a chance de ocorrer o evento será

$$\frac{p(x_i)}{1 - p(x_i)} = exp(\beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{ip}).$$

Observe que se adicionarmos um valor v, com $v \in \mathbb{R}$, em uma variável preditora contínua

qualquer do modelo, tomemos x_1, a chance do evento passa a ser

$$\begin{aligned}\frac{p(x_i)^*}{1-p(x_i)^*} &= exp(\beta_0+\beta_1(x_{1i}+v)+\ldots+\beta_p x_{ip}) \\ &= exp(\beta_0+\beta_1 x_{1i}+\ldots+\beta_p x_{ip}+v\beta_1) \\ &= exp(\beta_0+\beta_1 x_{1i}+\ldots+\beta_p x_{ip})exp(v\beta_1) \\ &= \frac{p(x_i)}{1-p(x_i)}exp(v\beta_1).\end{aligned}$$

Pode-se, então, calcular a razão de chances (*odds ratio* - OR) para a variável x_1+v com relação à variável x_1 de origem. Assim,

$$OR(x_1+v,x_1)=\frac{p(x_i)^*/1-p(x_i)^*}{p(x_i)/1-p(x_i)}=exp(v\beta_1).$$

O eventual resultado $exp(v\beta_1)$ naturalmente pode ser interpretado como a chance de ocorrência do evento nas observações que foram acrescidas com um determinado valor v na variável x_1.

Além disso, para fazer o ajuste de um modelo de regressão logística usa-se o método de máxima verossimilhança, dado por

$$\begin{aligned}L(y;(x,\beta)) &= \prod_{i=1}^{n}(P(Y_i=1|x_i,\beta))^{y_i}(1-P(Y_i|x_i,\beta))^{1-y_i} \\ &= \prod_{i=1}^{n}\left[\frac{exp(\alpha+\beta_0+\beta_1 x_{1i}+\ldots+\beta_p x_{pi})}{1+exp(\alpha+\beta_0+\beta_1 x_{1i}+\ldots+\beta_p x_{pi})}\right]^{y_i}\left[\frac{1}{1+exp(\alpha+\beta_0+\beta_1 x_{1i}+\ldots+\beta_p x_{pi})}\right]^{1-y_1}.\end{aligned}$$

Já para encontrar as estimativas dos coefientes β, basta maximizar $ln[L(y;(x,\beta))]$ por meio de algum algoritmo numérico iterativo, como o método de Escore de Fisher ou de *Newton-Raphson*. Uma vez estimando os β, isto é, encontrando os valores de $\left(\hat{\beta}_0+\ldots+\hat{\beta}_p\right)$, e substituindo-os na Equação (1), é possível obter as estimativas $\hat{p}(x_i)$. Se a probabilidade estimada for maior do que um certo limiar, o modelo retornará 1, caso contrário, zero. Um meio bastante prático para encontrar esse limiar é por meio da Curva ROC, vista no início desta seção.

O fato da Regressão Logística prover resultados em termos de probabilidades e ter um alto grau de confiabilidade, torna esse método bastante utilizado em problemas de classificação.

3.1.2 Análise Discriminante

Se na Regressão Logística modelava-se a distribuição da resposta Y dadas as covariáveis x_i, isto é, $f(Y_i=c|X=x_1,\ldots,x_p)$, com $c=\{0,1\}$, na Análise Discriminante a modelagem acontece sobre a distribuição das covariáveis x_i condicionada à resposta Y, isto é, $f(\mathbf{X}=x_1,\ldots,x_p|Y_i=c)$. Algumas vantagens do modelo de análise discriminante sobre o modelo de regressão logística estão relacionadas com a estabilidade: na Análise Discriminante os estimadores dos parâmetros não são afetados se as classes de respostas são bem definidas, assim como se o tamanho amostral n for pequeno e a distribuição das covariáveis seguir distribuição aproximadamente normal (James & Tibshirani, 2013). Podemos encontrar dois tipos: Análise Discriminante Linear e Análise Discriminante Quadrática. Vejamo-os a seguir.

3.1.2.1 Análise Discriminante Linear

Considere o caso em que temos dois grupos de classificação, C_0 e C_1.

Para cada distribuição $f(\mathbf{X}|Y_i = c)$ é suposto normalidade multivariada, assim como que suas respectivas matrizes de covariância sejam iguais, isto é,

$$\Sigma_{C_0} = \Sigma_{C_1} = \Sigma = E(\mathbf{X} - \mu)(\mathbf{X} - \mu)'.$$

Logo,

$$\mathbf{X} = x_1, \ldots, x_p | Y_i = c \sim N(\mu_c, \Sigma).$$

Em consequência, a função densidade de probabilidade de cada distribuição é

$$f(\mathbf{X}|Y_i = c) = \frac{1}{\sqrt{(2\pi)^p |\Sigma|}} e^{-(\mathbf{x}-\mu_c)'\Sigma^{-1}(\mathbf{x}-\mu_c)}.$$

Em um contexto geral, na Análise Discriminante Linear, espera-se encontrar a combinação linear $Y = \mathbf{L}'\mathbf{X}$, com Y sendo uma variável aleatória univariada e $\mathbf{L} = (l_1, \ldots, l_p)$ sendo um vetor discriminante de dimensão $p \times 1$, que maximiza a distância ao quadrado entre as médias populacionais univariadas de Y resultantes dos valores de $\mathbf{X} = x_1, \ldots, x_p$ (Morettin, 2020).

Como, neste caso, a classificação está entre dois grupos, as médias em questão são

$$\mu_{C_0 Y} = E(Y|C_0) = \mathbf{L}'\mu_\mathbf{0} \quad e \quad \mu_{C_1 Y} = E(Y|C_1) = \mathbf{L}'\mu_\mathbf{1},$$

em que $\mu_0 = E(\mathbf{X}|C_0)$ e $\mu_1 = E(\mathbf{X}|C_1)$.

A variância da variável dependente Y, igual em C_0 e C_1, é dada por

$$\sigma_Y^2 = Var(\mathbf{L}'\mathbf{X}) = \mathbf{L}'\Sigma\mathbf{L}.$$

Assim, vamos ter que a diferença ao quadrado entre as médias $\mu_{C_0 Y}$ e $\mu_{C_1 Y}$ em relação à variação dos valores de Y será

$$\begin{aligned} \frac{(\mu_{C_0 Y} - \mu_{C_1 Y})^2}{\sigma_Y^2} &= \frac{(\mathbf{L}'\mu_\mathbf{0} - \mathbf{L}'\mu_\mathbf{1})^2}{\mathbf{L}'\Sigma\mathbf{L}} \\ &= \frac{\mathbf{L}'(\mu_\mathbf{0} - \mu_\mathbf{1})(\mu_\mathbf{0} - \mu_\mathbf{1})'\mathbf{L}}{\mathbf{L}'\Sigma\mathbf{L}} \\ &= \frac{[\mathbf{L}'(\mu_\mathbf{0} - \mu_\mathbf{1})]^2}{\mathbf{L}'\Sigma\mathbf{L}}. \end{aligned} \tag{2}$$

Para qualquer $k \neq 0$, a Equação (2) é maximizada se $\mathbf{L} = k\Sigma^{-1}(\mu_\mathbf{0} - \mu_\mathbf{1})$. Quando $k = 1$, temos a chamada Função Discriminante Linear de Fisher, responsável por separar as observações perfeitamente entre os grupos de classificação C_0 e C_1.

Formalmente, ela é definida como sendo

$$Y_L = \mathbf{L}'\mathbf{X} = (\mu_\mathbf{0} - \mu_\mathbf{1})\Sigma^{-1}\mathbf{X}, \tag{3}$$

em que o resultado dessa função para qualquer observação x_0 é

$$y_{L_0} = (\mu_\mathbf{0} - \mu_\mathbf{1})\Sigma^{-1}\mathbf{x_0}.$$

Para construir a regra de classificação, deve-se, primeiro, encontrar o ponto médio m entre as duas médias μ_{C_0Y} e μ_{C_1Y} de modo que m funcione como o limiar para o qual uma nova observação será classificada em C_0 ou C_1. Matematicamente,

$$m = \frac{\mu_{C_0Y} + \mu_{C_1Y}}{2} = \frac{\mathbf{L}'\mu_\mathbf{0} + \mathbf{L}'\mu_\mathbf{1}}{2},$$

que pode ser reescrito como

$$m = \frac{(\mu_{\mathbf{C_0Y}} - \mu_{\mathbf{C_1Y}})'\Sigma^{-1}(\mu_\mathbf{0} - \mu_\mathbf{1})}{2}.$$

Portanto, com base na função discriminante e no ponto médio entre as médias univariadas, é decidido que

$$\begin{cases} x_0 & \text{será classificado em } C_0 \quad \text{se } y_{L_0} \geq m; \\ x_0 & \text{será classificado em } C_1 \quad \text{se } y_{L_0} < m. \end{cases}$$

No entanto, sabemos que, geralmente, as médias e as matrizes de covariância populacionais são desconhecidas, e por isso elas devem ser estimadas. Das amostras extraídas de C_0 e C_1, que são configuradas por uma matriz $p \times n_k$, n_k o número de observações em cada categoria, obtemos as estimatimativas $\bar{x}_0$ e $\bar{x}_1$ das médias e S_{C_0} e S_{C_1} das matrizes de covariância. A matriz de covariância, igual em C_0 e C_1, pode ser estimada por meio do estimador não-viciado

$$S_p = \frac{(n_0 - 1)S_{C_0} + (n_1 - 1)S_{C_1}}{n_0 + n_1 + 2}.$$

Logo, a função (3) é adaptada para

$$\hat{Y}_L = \hat{\mathbf{L}}'\mathbf{x} = (\bar{\mathbf{x}}_\mathbf{0} - \bar{\mathbf{x}}_\mathbf{1})'\mathbf{S}_\mathbf{p}^{-\mathbf{1}}\mathbf{x}.$$

E a nova regra de classificação passa a ser

$$\begin{cases} x_0 & \text{será alocado em } C_0 \quad \text{se } \hat{y}_{L_0} \geq \hat{m}; \\ x_0 & \text{será alocado em } C_1 \quad \text{se } \hat{y}_{L_0} < \hat{m}, \end{cases}$$

em que $\hat{y}_{L_0}$ é a estimativa da função linear $\hat{Y}_L$ e $\hat{m} = (\bar{x}_0 + \bar{x}_1)^\top S_p^{-1}(\bar{x}_1 + \bar{x}_2)$ é o ponto médio entre as médias estimadas.

Apesar disso, usar o ponto médio como limiar discriminante pode não ser uma boa ideia. Em James & Tibshirani (2013) é descrito um problema em que se pretende discriminar clientes em adimplentes e inadimplentes. Sob $\alpha = 0,5$, assim como acontece quando da aplicação do algoritmo *Naïve Bayes*, a taxa de erro geral é minimizada enquanto a taxa de erro dos indivíduos que realmente são inadimplentes é consideravelmente alta. Por essa razão, a Curva ROC é sugerida como métrica para encontrar o melhor limiar que atenuará a taxa de erro tanto do grupo de clientes inadimplentes como do grupo de clientes adimplentes.

3.1.2.2 Análise Discriminante Quadrática

Aqui, a suposição de que as distribuições $f(\mathbf{X}|Y_i = c)$ seguem uma distribuição normal multivariada também é feita. Entretanto, no modelo de análise discriminante quadrática as matrizes

de covariância podem diferir umas das outras. Assim,

$$\Sigma_{C_k} = E(\mathbf{X}-\mu)(\mathbf{X}-\mu)', \quad k = 0,1.$$

Em particular,

$$\mathbf{X} = x_1,\ldots,x_p | Y_i = c \sim N(\mu_c, \Sigma_c).$$

Portanto, a função de probabilidade de cada distribuição é dada por

$$f(\mathbf{X}|Y_i = c) = \frac{1}{\sqrt{(2\pi)^p |\Sigma_c|}} e^{-(\mathbf{x}-\mu_c)'\Sigma_c^{-1}(\mathbf{x}-\mu_c)}.$$

Normalmente, quando as matrizes de covariância dos grupos de classificação C_0 e C_1 são diferentes de forma significativa, as novas observações são classificadas de acordo com o grupo em que há maior variabilidade, ou seja, que é mais heterogêneo. Para medir a condição de heterogeneidade das matrizes de covariância, uma das técnicas que pode ser utilizada é o Teste M de Box. Uma vez que a suposição de heterogeneidade entre as matrizes de covariância de C_0 e C_1 não for violada, isto é, $\Sigma_{C_0} \neq \Sigma_{C_1}$ (de maneira significativa), a Análise Discriminante Quadrática passa a ser aplicável. A função discriminante quadrática é definida por

$$Y_Q = [(x-\mu_0)'\Sigma_{C_0}^{-1}(x-\mu_0) - (x-\mu_1)'\Sigma_{C_1}^{-1}(x-\mu_1)] + [ln|\Sigma_{C_0}| - ln|\Sigma_{C_1}|].$$

A regra de classificação é análoga à Análise Discriminante Linear, assim como o processo para a estimação dos parâmetros.

3.1.3 K-Vizinhos mais Póximos

É também conhecido como *k-nearest neighbours* ou KNN, em inglês (Izbicki, 2020).

Esse método, que é um dos mais usados no meio de *Machine Learning*, consiste numa ideia bastante simples: estimar a probabilidade condicional dos k-vizinhos mais próximos de uma determinada observação de teste x_0 (Izbicki, 2020), isto é,

$$P(Y_i = c | X = x_0), \quad c = \{0,1\}.$$

Em Morettin (2020) está exposto o algoritmo de predição KNN. A saber.

1. fixe um valor para k e uma observação de teste x_0.
2. no grupo de treinamento, identifique os k pontos que estejam mais próximos de x_0 segundo algum critério de distância (como exemplo, a Distância Euclidiana), e crie um conjunto W com esses pontos.
3. estime a probabilidade condicional da observação x_0 pertencer à classe C como sendo a fração dos pontos identificados em (2) que têm seus valores de Y iguais à C, isto é, $P(Y = c|x_0) = \frac{1}{k}\sum_{i\in W} I(y_i = c)$.

4. classifique x_0 com a classe que resultar em maior probabilidade.

É importante ressaltar que se deve ter critério na escolha de um valor para *k*, uma vez que valores altos de *k* podem restringir um modelo a ter estimadores com viés alto e baixa variabilidade, e valores baixos de *k* podem acarretar em um modelo com estimadores de características contrárias (Izbicki, 2020). Uma maneira de se escolher um valor para k, que também é conhecido como *tuning parameter k*, é por meio de algum método de validação cruzada (por exemplo, o método de validação cruzada *k-fold*).

A grande vantagem deste método está na sua simplicidade teórica e sobretudo de implementação, além da sua capacidade em aprender funções complexas. Em contrapartida, é bastante custoso computacionalmente e sensível a ruídos de covariáveis.

3.1.4 Support Vector Machines (SVM)

Support Vector Machines (Algoritmos de Vetores de Suporte) também é um método usado em problemas de classificação, sendo este idealizado por Cortes e colaboradores em 1995 (Morettin, 2020). A principal prerrogativa para fazer o uso desses modelos é de não ser necessário que os dados de treinamento sejam perfeitamente separáveis, além de sua performance ser boa em espaços com muitas variáveis. Por outro lado, os resultados do SVM não são de fácil interpretação, principalmente à medida que o conjunto de dados aumenta em tamanho.

Um espaço com dimensão $p > 2$ é composto por um hiperplano (subespaço) de tamanho $p-1$ e definido por $f(x) = \alpha + \beta_1 X_1 + \ldots + \beta_p X_p = 0$. E de forma bastante geral, sabendo que cada região do hiperplano representa uma categoria, em um caso de classificação binária, uma observação x_i, dada por um ponto de coordenadas $(x_1, \ldots, x_p)$, estará localizada em um lado do hiperplano se $f(x) > 0$ e estará do outro lado, caso contrário.

A margem m define a menor distância entre o(s) hiperplano(s) e as observações mais próximas de cada classe, sendo que sobre essas observações é traçada uma linha, a qual é denominada fronteira. Na fronteira, encontram-se os vetores de suporte (*support vectors*), pontos considerados influentes diretos na construção do classificador.

A separação em classes distintas pode ocorrer por meio de fronteiras lineares e não-lineares. No caso em que as fronteiras sejam lineares, há duas situações: os dados podem ser separados de maneira linear e não-linear. Quando linearmente separáveis, um hiperplano maximiza sua distância com os vetores de suporte, sendo considerado um hiperplano de margem máxima. Por outro lado, um hiperplano dito com margem flexível minimiza os erros de classificação decorrentes da impraticidade de se classificar dados não linearmente separáveis (Figura 2) (Morettin, 2020).

Considere um problema de classificação no qual $\mathscr{X} \in \mathbb{R}^p$, com $\mathscr{X}$ sendo o espaço das observações, e que $Y \in \{0,1\}$, em que que $y = 0$ corresponde à classe C_0 e $y = 1$ denota a classe C_1. Partindo desse pressuposto, a seguir serão descritos com mais detalhes os dois tipos

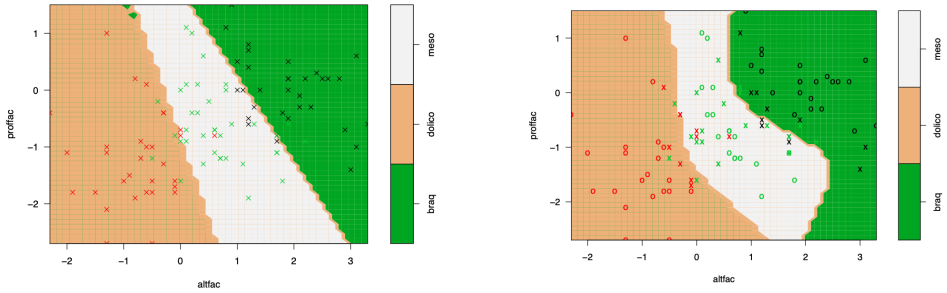

Figura 2: Exemplos de fronteiras de um (a) hiperplano de margem flexível e de (b) um hiperplano de margem não-linear. Fonte: Morettin e Singer (2020).

de fronteiras de separação em que o SVM se faz presente.

3.1.4.1 Fronteiras Lineares

Admitamos que as observações sejam totalmente separáveis linearmente por um hiperplano no conjunto de treinamento, segundo a classificação quando $y = 0$ ou quando $y = 1$. Como já visto anteriormente, esse hiperplano, ou separador, é conhecido como Classificador de Margem Máxima (CMM).

No caso em que há apenas duas variáveis, o CMM é uma reta, que por sua vez é definida por $\alpha + \beta_1 X_1 + \beta_2 X_2 = 0$. Logo, o plano é separado nas regiões $\alpha + \beta_1 X_1 + \beta_2 X_2 < 0$ e $\alpha + \beta_1 X_1 + \beta_2 X_2 > 0$.

Para o cenário em que hajam mais de duas variáveis, seja uma matriz com p variáveis, em que cada variável tenha n observações, isto é, $\mathbf{X}_{nxp}$. Em seguida, define-se o vetor de pesos que indica a direção perpendicular ao hiperplano por $\beta = (\beta_1, \ldots, \beta_p)^\top$, o vetor da j-ésima coluna de $\mathbf{X}$ por $x_j = (x_{1j}, \ldots, x_{nj})^\top$ e o vetor de teste por $x_{i_0} = (x_{i1_0}, \ldots, x_{ip_0})^\top$. O conjunto de treinamento é dado por $T = \{(x_i, y_i)\}$, com i representando o i-ésimo indivíduo nesse conjunto.

Vamos ter que uma função classificadora baseada em um hiperplano no espaço $\mathbb{R}^p$ será da forma

$$f(\mathbf{x}) = \alpha + \beta^\top \mathbf{x}_j. \tag{4}$$

Portanto, a condição

$$\begin{cases} \text{se } f(\mathbf{x}_{i_0}) = \alpha + \beta^\top \mathbf{x}_{i_0} < 0, & \text{a observação } \mathbf{x}_{i0} \text{ será classificada tal que } y_i = 0; \\ \text{se } f(\mathbf{x}_{i_0}) = \alpha + \beta^\top \mathbf{x}_{i_0} > 0, & \text{a observação } \mathbf{x}_{i0} \text{ será classificada tal que } y_i = 1 \end{cases}$$

descreve a regra de classificação atribuída à Equação (4).

Para qualquer ponto x_{i_0} corretamente classificado, $y_i f(x_{i_0}) > 0$, e a distância desse ponto ao hiperplano é dada por

$$m = \frac{|f(\mathbf{x})|}{||\beta||} = \frac{y_i(\alpha + \beta^\top \mathbf{x}_j)}{||\beta||}.$$

Caso hajam muitos hiperplanos que façam a separação das observações, escolhe-se aquele com a maior margem entre os pontos mais próximos das classes C_0 e C_1. Em outras palavras,

espera-se maximizar a distância do hiperplano aos vetores de suporte de modo que sejam escolhidos α e β que melhor satisfazem essa condição. Matematicamente,

$$arg\,\underset{\alpha,\beta}{max}\left\{\frac{1}{||\beta||}min_i[y_i(\alpha+\beta^\top \mathbf{x}_j)]\right\},$$

que pode ser resolvida usando o método de Multiplicadores de Lagrange (Morettin, 2020).

Não obstante, as classes podem ser inseparáveis linearmente. Quando isso acontece, é impossível que o hiperplano faça a separação sem cometer violações de classificação. Logo, é razoável que pontos se encontrem do lado errado da margem ou do hiperplano. Nesse sentido, o classificador de margem flexível (CMF) é usado para penalizar esses pontos, cada um com um peso de valor baixo, por meio das chamadas variáveis de folga, a fim de reduzir a indução ao erro no processo de classificação (Morettin, 2020).

Em particular, as variáveis de folga, que podem ser expressas como $\xi_i = (\xi_1, \ldots, \xi_n)^\top$, permitem que pontos estejam do lado incorreto da margem ou do hiperplano de tal forma que sejam classificados erroneamente. Quando isso ocorre, $\xi_i > 1$. Do contrário, quando $\xi_i < 1$, mas maior do que zero, significa que os pontos estão dentro dos limites da margem e do lado certo do hiperplano, ou seja, eles são classificados corretamente. O mesmo acontece quando $\xi_i = 0$, exceto pelos pontos estarem exatamente sobre a margem (Morettin, 2020).

Embora em cenários diferentes, o objetivo do CMF é o mesmo quando se considera um hiperplano com margem máxima: maximizar a margem do hiperplano, que agora é definida pelos parâmetros α, β e ξ. Para isso, basta minimizar

$$\frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n}\xi_i, \tag{5}$$

em que $C \in [0, \infty)$ é um parâmetro de custo que controla as variáveis de folga, ou seja, permite uma certa quantidade de classificações erradas desde que estas sejam penalizadas por um custo C.

Se o custo escolhido for baixo, a margem é menor, ocasionando um modelo com mais erros de classificação e com problemas de sobreajustamento em virtude da variância alta. Alternativamente, ao se escolher um custo mais alto, a penalização sobre os erros cometidos é maior e as margens se tornam mais suaves, generalizando melhor seus resultados. Quando C é exatamente igual a zero, significa que todos os pontos são classificados corretamente, pois $C = 0$ implica em $\xi_i = \ldots = \xi_n = 0$. O parâmetro de custo C pode ser determinado via método de validação cruzada (Morettin, 2020).

Minimizando a Equação (5) com o auxílio dos multiplicadores de Lagrange (Morettin, 2020), assegura-se a margem máxima tal como a minimização dos erros.

3.1.4.2 Fronteiras Não-Lineares

No mundo real, os dados dos quais dispomos quase sempre não se comportam de maneira linear, não sendo possível, portanto, a divisão linear entre classes. Nesses casos, utilizam-se *kernels* para aumentar a dimensão de um espaço $\mathscr{X}$, denominando-o espaço de características

$\mathscr{H}$, de tal forma que seja possível encontrar um hiperplano que separe os dados linearmente e ter uma classe de estimadores mais robustos.

Considere o espaço com produto interno $\mathscr{X}$ tal que $\langle x_i, y_i \rangle, \forall x_i, y_i \in \mathscr{X}$.

Seja

$$\phi : \mathscr{X} \to \mathscr{H}$$

a função característica que mapeia todas as observações do espaço de origem para um espaço de características $\mathscr{H}$. Assim, o produto interno nesse novo espaço é da forma $\langle \phi(x_i), \phi(y_i) \rangle$.

No entanto, é complexo calcular ϕ devido à alta dimensionalidade do espaço $\mathscr{H}$. Por conveniência, usa-se um *kernel* $K(x_i, y_i) = \langle \phi(x_i), \phi(y_i) \rangle$ tal qual

$$K : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$$

é a função para construir um hiperplano ou classificador de margem não-linear (CMNL).

Tal classificador depende somente dos pontos de vetores de suporte e consiste na combinação entre um CMF e algum *kernel* do tipo não-linear (polinomial, *splines*, gaussiano etc). Formalmente, se

$$CFM = \sum_{i \in S} \gamma \langle x, y \rangle + \delta,$$

com S sendo o conjunto de vetores suporte, γ um parâmetro para suavização da margem e δ um parâmetro de movimentação do hiperplano, então

$$CMNL = \alpha + CFM.$$

Em particular, o algoritmo de *support vector machines* polinomial opera com um *kernel* polinomial, que é dado por $K(x_i, y_i) = (c + x_i^\top y_i)^p$, em que c é uma constante que controla a influência de polinômios com diferentes graus.

3.1.5 Árvores

Leo Breiman foi quem criou, por volta de 1980, os modelos estatísticos baseados em árvores (Morettin, 2020).

Um espaço $\mathbb{R}$ é gerado por um vetor de covariáveis x_i e segmentado em subregiões que são pautadas por alguma medida de erro. Cada subregião representa um nó e de cada nó obtém-se um resultado final, conhecido como folha (Izbicki, 2020). É possível particionar uma árvore de decisão até que todos os valores ou categorias de todas as variáveis preditoras alcancem seu próprio nó, contudo isso ocasionaria um problema de sobreajuste.

A quantidade de nós determina o tamanho da árvore, e a profundidade ideal pode ser obtida por meio da técnica de poda. Podar uma árvore em ML é o mesmo que usar validação cruzada para avaliar o erro de predição no conjunto de validação de cada nó, optando-se pelos nós cujos erros são menores. Dessa forma, com menos subregiões é garantida a baixa variabilidade nas folhas.

A predição dos modelos baseados em árvores é feita segundo a média, em problema de regressão, ou a moda, em problema de classificação, da variável resposta Y. Quando combinadas inúmeras árvores, tem-se as florestas aleatórias (*random forests*), que nada mais são do que generalizações desses modelos baseados em árvores.

As árvores de decisão possuem vantagens como serem de fácil implementação computacional e produzirem resultados satisfatoriamente interpretáveis, ainda que tenham baixa performance preditiva se comparadas a modelos mais complexos.

Neste estudo, serão discutidos os modelos de árvores e generalizações voltados para o contexto de classificação.

3.1.5.1 Árvores de Classificação

Sejam os dados de treinamento

$$(x_i, y_i) \in \mathbb{R}^p \times (1, \ldots, n),$$

em que p é a quantidade de variáveis explicativas do modelo.

As árvores de classificação são usadas quando a variável resposta Y é de natureza categórica, ou seja, qualitativa. Intuitivamente, o valor (no caso, a classe) predito para a resposta Y será dado pela moda das observações de treinamento que pertencem a uma determinada subregião R_j. Ou seja, a função $g(x)$ que satisfaz essa condição é

$$g(x) = moda\{y_i : x_i \in R_j\}, \quad j = 1, \ldots, m.$$

De outro modo, uma árvore de classificação determinará a subregião R_j para uma observação x_i, que por sua vez será classificada com a classe mais frequente entre os dados de treinamento que pertencem à mesma subregião.

Espera-se que essas partições resultem em predições em que os erros sejam minimizados. A taxa de erro de classificação (TEC) é tida como uma boa medida de erro quando se tem um problema de classificação, assim como entropia ou Índice de Gini (Morettin, 2020).

3.1.5.2 *Bagging* e Florestas Aleatórias

Modelos *Bagging*, também conhecidos como agregação *bootstrap*, e Florestas Aleatórias são generalizações de árvores de decisão que visam diminuir a variância das predições, sejam em problemas de regressão ou de classificação. Em seus contextos gerais, são feitas combinações de resultados de árvores de decisão ajustadas a fim de aumentar o poder preditivo em relação a se fosse feito sob uma única árvore.

Como estamos considerando um problema de classificação, suponha o seguinte cenário em que a variável resposta Y seja qualitativa.

Em *bagging*, utiliza-se o método de reamostragem *bootstrap* para gerar B amostras de *bootstrapping*[1], das quais é obtida uma árvore (classificador) após feito o ajuste do mesmo modelo em

[1] Breiman (1996) sugere que 25 amostras de *bootstrapping* é o ideal para o bom funcionamento deste método.

cada uma delas. A função de predição, ou seja, do classificador final, será dada pela combinação da b-ésima árvore de classificação, e a classe predominante dentre elas será a escolhida para esse classificador. Formalmente,

$$g(x) = moda\{g^{(b)}(x), b = 1, \ldots, B\}.$$

É importante dizer que em amostras selecionadas via método *bootstrap* é comum que as árvores ajustadas tenham predições parecidas, pois uma dada observação x_i pode estar presente mais de uma vez na amostra. Isso faz com que as funções de predição do modelo sejam fortemente correlacionáveis.

Uma forma de contornar esse problema é usar florestas aleatórias, uma aplicação da técnica *bagging* sobre árvores de decisão capaz de controlar o grau de correlação no processo de criação das árvores a partir da seleção de m dentre as p covariáveis existentes em cada ajuste feito. Usualmente, define-se m como sendo $\sqrt{p}$ covariáveis, entretanto o valor de m pode ser determinado por validação cruzada (Amorim, 2019).

Um algoritmo que apresenta como o modelo de florestas aleatórias opera está em Amorim (2019) e pode ser visto a seguir.

1. seleciona-se, aleatoriamente, m dentre as p covariáveis; $m \leq p$.
2. gera-se uma amostra de *bootstrapping* a partir da amostra original.
3. ajusta-se uma árvore de decisão usando o subconjunto de m covariáveis do passo (1) e a amostra do passo (2).

Em cada iteração do algoritmo é obtida uma classe predita, e ao final de M[2] iterações, a classe com maior ocorrência será a predição final para a observação.

3.1.5.3 *Boosting/XGBoost*

O método de *boosting* se cerca da mesma ideia de *bagging* e florestas aleatórias, diferenciando-se somente quanto à maneira em que são treinados. Em *boosting* também é feita a combinação entre diversas árvores de decisão com a intenção de reduzir a variância dos resultados e, consequentemente, aumentar sua precisão. Porém, a maneira com que ocorre a combinação dessas árvores é diferente.

Em geral, no algoritmo *boosting* as árvores são combinadas de forma sequencial, ou seja, cada árvore $\hat{f}^i(x)$ é construída a partir dos resíduos de uma árvore $\hat{f}^{i-1}(x)$ previamente ajustada - não sendo necessária, portanto, a utilização de amostras de *bootstrapping*, como em *bagging*.

Cada função $\hat{f}^i(x)$ adicionada ao modelo tem a estrutura de uma árvore, e mesmo que tenham baixa profundidade (poucos nós), ajustar todas elas em uma única vez seria muito mais difícil. Aprendê-las individualmente de tal forma que uma nova árvore é acrescida à medida que os

[2] Aproximadamente 200 iterações são adequadas para conseguir resultados satisfatórios, como recomenda Amorim (2019).

resíduos são atualizados garante a boa performance do *boosting*, embora esse seja o mesmo motivo que o faz ser considerado um algoritmo de aprendizagem lenta.

Seja $\hat{f}(x)$ a função de predição estimada para o modelo. Em cada iteração i, uma nova árvore $\hat{f}^i(x)$ é somada à $\hat{f}(x)$ ponderada por um parâmetro de encolhimento $\lambda > 0$ responsável por controlar a velocidade com que o algoritmo aprende. Então, temos que

$$\hat{f}^{(0)}(x) = 0, \quad i = 0$$

$$\hat{f}^{(1)}(x) = \hat{f}^{(0)}(x) + \lambda \hat{f}^{(1)}(x), \quad i = 1,$$

$$\hat{f}^{(2)}(x) = \hat{f}^{(1)}(x) + \lambda \hat{f}^{(2)}(x), \quad i = 2,$$

$$\vdots$$

$$\hat{f}^{(n)}(x) = \sum_{k=1}^{n} f_k(x) = \hat{f}^{(n-1)}(x) + \lambda \hat{f}^{(n)}(x), \quad i = n.$$

Como resultado, a cada iteração, $\hat{f}(x)$ torna-se cada vez mais robusta em termos de predição.

Os principais hiperparâmetros, a saber λ, o número de árvores e o tamanho de cada uma delas, podem ser obtidos por meio do método de validação cruzada (Amorim, 2019).

Há inúmeras variações de *boosting* e sua implementação. Consideramos, aqui, o *XGBoost* (*Extreme Gradient Boosting*).

O *XGBoost* é uma aplicação do método de *Gradient Boosting*, cuja função é minimizar uma função de perda específica $L(y, f(x))$ por meio do algoritmo de gradiente descendente (Friedman, 2001). Em cada iteração, uma nova árvore é ajustada utilizando o gradiente da função de perda da árvore anterior, isto é,

$$g_i = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=f^{i-1}(x)}, \quad i = 1, \ldots, n.$$

Assim, a função de perda é minimizada gradativa e progressivamente ao longo das iterações.

De modo complementar, o objetivo do *gradient boosting* é obter uma estimativa $\hat{f}(x)$ para

$$f^*(x) = arg\min_f E_{y,x}[L(y, f(x))] = arg\min_f E_x[E_y(L(y, f(x)))|x]$$

que minimiza o valor esperado da função de perda $L(y, f(x))$.

3.2 Modelos de interpretabilidade

3.2.1 Gráfico de Dependência Parcial

O gráfico de dependência parcial é a representação gráfica do efeito marginal que uma ou até duas covariáveis têm sobre o resultado predito de um modelo de ML (Friedman, 2001).

A partir de um modelo ajustado não interpretável $f(\mathbf{X}, \mathbf{C})$, em que $\mathbf{X}$ é a matriz de covariáveis de interesse e $\mathbf{C}$ é a matriz das covariáveis restantes, o algoritmo inerente a este gráfico compreende os seguintes passos:

1. fixe valores para $\mathbf{X}$.

2. para cada valor x_i, $i = 1, \ldots, M$, fixado:

   2.1 substitua o valor observado de $\mathbf{X}$ por x_i em cada uma das n observações da amostra e calcule o valor predito do modelo, i.e., $\hat{f}(x_i, c)$.

   2.2 calcule a média das n predições:

$$\bar{f}_{x_i}(x_i) = \frac{1}{n} \sum_{j=1}^{n} \hat{f}(x_i, c_j). \tag{6}$$

3. construa o gráfico dos valores x_i contra as médias das predições $\bar{f}(x_i)$.

Em resumo, a função de dependência parcial (DP) dada na Equação (6) calcula o efeito médio das covariáveis de $\mathbf{X}$ ao marginalizar a distribuição das predições sobre as covariáveis para as quais não há interesse, pois, dessa forma, a função se torna dependente apenas das covariáveis de $\mathbf{X}$.

A grande vantagem desse gráfico está em sua interpretação, que é intuitiva e causal. Nele, observamos qual é o efeito médio na predição para diferentes valores das covariáveis de $\mathbf{X}$. No entanto, essa interpretação só é coerente quando os preditores $\mathbf{X}$ e $\mathbf{C}$ não são correlacionados. Em caso de se haver correlação, a média pode ser influenciada por valores que não fazem sentido para todas as observações, podendo causar interpretações pouco prováveis ou mesmo irreais.

3.2.2 Gráfico da Esperança Condicional Individual

Enquanto gráficos de dependência parcial exibem a relação média de uma variável com o resultado predito, em gráficos da esperança condicional individual o grau de dependência entre variável e predição é considerado individualmente, isto é, para cada observação.

De outro modo, a curva de um gráfico de dependência parcial é exatamente a média das curvas de um gráfico da esperança condicional individual. Assim, o algoritmo desse gráfico se restringe aos passos (1) e (2.2.1) do algoritmo do gráfico DP, em que cada umas das n curvas é formada pelo ponto

$$\left(x_i, \hat{f}(x_i, c_j)\right), i = 1, \ldots, M,\ j = 1, \ldots, n.$$

Sendo possível quantificar as relações separadamente, a interpretação é ainda mais direta e as relações heterogêneas de covariáveis são mais facilmente identificáveis. Apesar disso, o problema decorrente de uma possível correlação entre a variável sob investigação e as demais é o mesmo dos gráficos de dependência parcial.

3.2.3 Gráfico de Efeitos Locais Acumulados

Tem o mesmo objetivo do gráfico de dependência parcial: calcular o efeito médio das covariáveis sobre a predição de um modelo de ML. No entanto, o método dos efeitos locais acumulados atua sobre a distribuição condicional das covariáveis, acumulando em uma grade a média

das diferenças entre predições condicionadas a pequenas variações nos valores de um dado x do preditor $\mathbf{X}$. Por essa razão, o gráfico de efeitos locais acumulados (ELA) é uma solução para o gráfico de dependência parcial, pois o efeito estimado para as covariáveis de $\mathbf{X}$ não é interferido por valores de variáveis correlacionadas.

A partir de uma matriz $\mathbf{X}$ de interesse, a construção do gráfico de efeitos locais acumulados se dá pelo seguinte algoritmo:

1. divida $\mathbf{X}$ em M intervalos.
2. calcule os efeitos locais para cada uma das m_i observações dentro do i-ésimo intervalo:

$$\hat{f}_{d_{ij}} = (x_i^+ - c_j) - \hat{f}(x_i^- - c_j),\ j = 1, \ldots, m_i,$$

   com x_i^+ e x_i^- correspondendo, nesta ordem, aos limites superior e inferior do intervalo i.
3. calcule a média acumulada para cada valor x de $\mathbf{X}$:

$$\bar{f}_a(x) = \sum_{i=1}^{k(x)} \frac{1}{m_i} \sum_{j=1}^{m_i} \hat{f}_{d_{ij}},$$

   em que $k(x)$ é o índice do intervalo que x faz parte.
4. calcule o valor centralizado de $\bar{f}_a(x)$ para todas as n observações:

$$\hat{f}_c(x) = \hat{f}_a(x) - \frac{1}{n} \sum_{i=1}^{n} \hat{f}_a(x_i)$$

No passo (2), os efeitos são considerados locais porque a diferença entre predições significa o quanto o resultado predito é influenciado por um valor de x de $\mathbf{X}$ e em torno próximo de x. No passo (3), o efeito médio do preditor $\mathbf{X}$ é calculado ao somar a diferença média das predições de cada intervalo. No passo (4), os resultados centrados em zero permitem interpretar o valor de um ponto em uma curva gerada pelo gráfico de efeitos locais acumulados como a diferença para o valor predito médio.

Além disso, se o interesse está em investigar o efeito combinado de duas covariáveis, é preferível utilizar gráficos de dependência parcial a gráficos de efeitos locais acumulados - desde que não haja alguma correlação significativa das covariáveis de interesse com as outras do modelo. Para obter essa informação usando gráficos de efeitos locais acumulados seriam necessários um gráfico ELA para cada uma das covariáveis e para as duas covariáveis juntas e saber o resultado da predição média total. De forma geral, com mapas de calor ELA é interessante descobrir e explorar as relações entre duas covariáveis, mas não seu efeito total, já que se trata apenas do efeito da interação entre elas.

3.2.4 Interação das Covariáveis

Essa abordagem avalia o peso da interação de dada covariável com as demais, conhecida como interação bidirecional, e da interação entre todos os pares possíveis de covariáveis, que

reflete a interação total. Uma maneira de estimar a força da interação é medir o quanto da variação da predição depende da interação das covariáveis. Uma medida possível é a estatística H, introduzida por Friedman et al. (2008).

A variância explicada de uma interação é determinada pela diferença entre a função de dependência parcial observada e a função de dependência parcial sem interação.

Em particular, se não há interação entre duas covariáveis, digamos j e k, a função DP definida na Equação (6) pode ser decomposta em

$$DP_{jk}(x_j, x_k) = DP_j(x_j) + DP_k(x_k),$$

em que $DP_j(x_j)$ e $DP_k(x_k)$ são as funções DP isoladas das covariáveis j e k.

Agora, se a covariável j não interagir com nenhuma outra, a função DP sem interação nessa situação é dada por

$$\hat{f}(x) = DP_j(x_j) + DP_{-j}(x_{-j}),$$

com $DP_{-j}(x_{-j})$ sendo a função DP de todas as covariáveis, exceto j.

Portanto, a estatística H usada para medir a força de uma interação bidirecional é

$$H_j^2 = \frac{\sum_{i=1}^{n} \left[DP_{jk}\left(x_j^{(i)}, x_k^{(i)}\right) - DP_j\left(x_j^{(i)}\right) - DP_k\left(x_k^{(i)}\right) \right]^2}{\sum_{i=1}^{n} DP_{jk}^2\left(x_j^{(i)}, x_k^{(i)}\right)}.$$

E quando a interação é total, a estatística é

$$H_j^2 = \frac{\sum_{i=1}^{n} \left[\hat{f}\left(x^{(i)}\right) - DP_j\left(x_j^{(i)}\right) - DP_{-j}\left(x_{-j}^{(i)}\right) \right]^2}{\sum_{i=1}^{n} \hat{f}^2\left(x^{(i)}\right)}.$$

Se H resultar em zero, significa que não havia qualquer interação. Por outro lado, se o resultado for 1, quer dizer que toda a variância pode ser explicada pela função DP sem interação, seja DP_{jk} ou $\hat{f}$. Em um cenário de interação bidirecional, o resultado $H_j^2 = 1$ expressa a função DP como uma constante e que o efeito sobre as decisões do modelo é ocasionado exclusivamente pela interação entre as covariáveis j e k.

Ao usar todos os n pontos de dados, a estatística de interação H requer muita carga computacional, e uma das formas de contornar essa problemática é selecionar uma parte de n. No entanto, isso aumenta a variabilidade das estimativas DP, tornando os resultados da estatística inconsistentes. Além disso, H é uma medida que funciona bem sobre variáveis independentes. Do contrário, as interpretações vão decorrer de interações inverossímeis.

Ainda assim, a estatística H é uma ferramenta poderosa em detectar quaisquer tipos de interações e por ser comparável entre covariáveis e tipos de modelo, além de não estar limitada a uma interação bidirecional somente (ela também pode ser da forma tridirecional ou com mais variáveis).

3.2.5 Importância da Covariável por Permutação

É um método limitado a modelos de ML não-supervisionados e a ideia geral da teoria que há por detrás da importância da covariável por permutação, introduzida por Breiman (2001), diz que a importância de uma covariável em um modelo é quantificada pelo aumento do erro em sua predição após permutar essa variável. Se permutando os valores da covariável o erro da predição aumenta, então essa variável é importante para o modelo. Caso contrário, isto é, se o erro da predição não se altera, a variável é insignificante.

Assim como a teoria, o algoritmo que mede a importância da covariável por permutação é bastante simples. Dados como entrada o modelo treinado f, a matriz de preditores $\mathbf{X}$, o vetor da variável resposta y e uma função de perda $L(y, f(x))$, o algoritmo proposto por Fisher et al. (2018) é dado por:

1. estime o erro do modelo por meio da função de perda $L(y, f(\mathbf{X}))$ (como exemplos, o erro quadrático médio em problemas de regressão e a taxa de erro de classificação em problemas de classificação).

2. para cada j-ésima variável, $j = 1, \ldots, p$:

   2.1 gere a matriz de variáveis permutadas $\mathbf{X_p}$ ao permutar a variável j nos dados da matriz original $\mathbf{X}$.

   2.2 estime o erro do modelo com base na matriz de variáveis permutadas $\mathbf{X_p}$, i.e., $e_p = L(y, f(\mathbf{X_p}))$.

   2.3 calcule a importância da covariável j por permutação:

   $$I_j = e_p / e_{mod} \quad \text{ou} \quad I_j = e_p - e_{mod}.$$

3. ordene as importâncias I de forma decrescente.

O passo (2.2.1) do algoritmo bloqueia a relação entre a covariável j e o verdadeiro valor de y, permitindo que o efeito da covariável e os efeitos da interação dessa covariável com as demais sejam considerados na performance do modelo.

Um grande ganho ao avaliar a importância de uma covariável por permutação é não precisar reciclar o modelo, que, como sabemos, pode ser um processo custoso e demorado computacionalmente. Ao permutar as covariáveis, ganha-se, no mínimo, tempo para identificar quais delas são, de fato, relevantes. Apesar disso, a importância da covariável por permutação é calculada com base em uma estimativa de erro do modelo, o que não é interessante se o objetivo maior é saber o quanto de variância pode ser explicada por cada covariável em vez de querer examinar se o desempenho do modelo diminui permutando os valores de uma covariável.

Em Molnar (2019), levanta-se a questão sobre em qual amostra deve-se calcular a importância da covariável, se na amostra de treino ou na de teste. E ao que parece, por enquanto, fica a critério de cada usuário. Se o objetivo é saber o quanto o modelo é dependente de cada variável,

opta-se pelos dados de treinamento. Mas se o desejo está em medir a contribuição de cada variável sobre as decisões acertadas do modelo em dados novos, é sugerido escolher os dados de teste.

3.2.6 Modelo Interpretável Substituto Global

Um modelo substituto global é um modelo interpretável treinado para aproximar as predições de um modelo caixa-preta. Dessa forma, é possível tirar conclusões sobre o modelo caixa-preta interpretando globalmente o modelo substituto, que usa a predição do modelo não-interpretável como sua variável resposta.

De outro modo, ao usar um modelo substituto global, o objetivo está em aproximar a função de predição do modelo caixa-preta f à função de predição do modelo substituto g, com a diferença de que g deve ser interpretável. A escolha do modelo substituto independe do modelo caixa-preta que está sendo usado, uma vez que não é necessário conhecer o funcionamento interno do modelo caixa-preta.

O algoritmo para obter um modelo interpretável substituto global consiste nos seguintes passos:

1. selecione o conjunto ou um subconjunto de dados X que foi usado para treinar o modelo caixa-preta f.
2. para este conjunto ou subconjunto de dados X, obtenha as predições de f.
3. ajuste um modelo interpretável g com os dados selecionados em (1) e as predições obtidas em (2).
4. avalie o quão bem o modelo simples g replica as predições do modelo complexo f.
5. interprete g e tire as conclusões acerca de f.

Uma medida recomendada e bastante usada no passo (4) é a R-quadrado, dada por

$$R^2 = 1 - \frac{SQE}{SQT} = 1 - \frac{\sum_{i=1}^{n} \left(\hat{y}_i^* - \hat{y}_i\right)^2}{\sum_{i=1}^{n} \left(\hat{y}_i - \tilde{\hat{y}}\right)^2},$$

em que $\hat{y}_i^*$ é a predição da i-ésima observação do modelo substituto e $\hat{y}_i$ é a predição do modelo caixa-preta.

Essa medida, que extrai de 1 a razão entre a soma de quadrados do erro (SQE) e a soma de quadrados total (SQT), pode ser interpretada como o percentual de variância explicada pelo modelo substituto global. Se $R^2 \approx 1$, significa que o modelo substituto interpretável g se comporta de maneira similar ao modelo caixa-preta f, possibilitando generalizar as interpretações de g para f. Do contrário, se $R^2 \approx 0$, é inviável interpretar o modelo caixa-preta a partir do modelo substituto simples.

A principal vantagem deste método de interpretabilidade é poder escolher qualquer modelo que seja interpretável, sendo possível optar por aquele em que as interpretações são mais familiares para o usuário. Por outro lado, como o modelo substituto não tem acesso aos resultados reais, as interpretações feitas se referem exclusivamente ao modelo caixa-preta, e, portanto, não são baseadas nos dados.

3.2.7 Modelo Interpretável Substituto Local

Ribeiro et al. (2016) propõem uma implementação do modelo interpretável substituto local (*Local Interpretable Model-Agnostic Explanations* - LIME, em inglês) para aproximar as predições do modelo caixa-preta. A ideia é a mesma do método do substituto global visto na seção anterior. Mas ao invés de treinar um modelo cuja interpretação é verdadeira globalmente, o LIME se concentra em explicar as predições individualmente, ou seja, para cada observação é possível avaliar o quanto cada covariável influenciou na sua predição.

De forma geral, o LIME funciona como um modelo interpretável simples que se aproxima bem do modelo caixa-preta nas proximidades de uma observação x de interesse, gerando, portanto, uma interpretação que seja verdadeira apenas em torno da observação que se quer explicar. Sendo assim, as predições feitas pelo modelo caixa-preta podem facilmente ser interpretadas de forma individual pelo modelo interpretável escolhido.

O algoritmo associado a essa técnica tem como sequência de passos:

1. para cada predição do modelo caixa-preta f a ser explicada, permute as observações n vezes.
2. obtenha a predição de cada observação permutada por meio do modelo f.
3. pondere as observações permutadas segundo sua proximidade com a observação original ao calcular uma medida de distância e dissimilaridade.
4. selecione as m variáveis consideradas mais importantes para as predições feitas por f e use-as para explicar os dados permutados.
5. ajuste um modelo interpretável g ao conjunto de dados permutados, em que as m variáveis selecionadas em (4) sejam as covariáveis e as predições do modelo caixa-preta sejam a variável resposta.
6. interprete localmente um ponto x de f com base nas estimativas de g geradas pela minimização de uma função de perda L somada a uma medida de complexidade Ω:

$$\xi(x) = arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g), \tag{7}$$

em que π_x define o tamanho da vizinhança em torno de x e G é o conjunto de modelos interpretáveis possíveis.

É possível ver na Equação (7) que as explicações produzidas pelo estimador LIME vão depender dos valores escolhidos para seus parâmetros, em especial o tamanho da vizinhança. Entretanto, precisar pré-definir esses valores é considerado um dos principais pontos negativos deste método, uma vez que é necessário testar diferentes configurações até que as interpretações façam sentido para um determinado contexto. Outra grande fraqueza do LIME é apontada por Alvarez-Melis & Jaakkola (2018), que mostram em cenários simulados que as explicações são instáveis mesmo para dois pontos de dado muito próximos. Já Amorim (2019) discute a dificuldade em se encontrar um modelo interpretável que explique bem localmente as predições de modelos muito complexos, principalmente de regressão e que tenham muitas covariáveis. Mas ainda assim, o LIME é ainda um dos poucos métodos de interpretabilidade aplicável a dados de natureza tabular, textual e de imagem.

3.2.8 Valores Shapley

Desenvolvido por Shapley (2016), o valor Shapley é um conceito da Teoria dos Jogos que descreve como distribuir, de forma justa, uma premiação aos jogadores de uma coalizão de acordo com a contribuição individual de cada um no resultado final de um jogo. Essa contribuição, o valor Shapley, é medida pela média da diferença entre todas as contribuições marginais de todas as coalizões possíveis que contém e não contém determinado jogador.

Em *Machine Learning*, podemos traduzir o "jogo" como o modelo preditivo, os "jogadores" como as covariáveis e o "prêmio" como a predição. Por sua vez, o valor Shapley é definido pela média ponderada (por uma constante normalizadora) da soma de todas as diferenças possíveis entre os modelos treinados com $\left(f_{S\cup\{j\}}\right)$ e sem (f_S) a j-ésima covariável sob investigação. Formalmente,

$$\phi_j = \sum_{S\subseteq N\setminus\{j\}} \frac{|S|!(|N|-|S|-1)}{N!}\left[f_{S\cup\{j\}}\left(x_{S\cup\{j\}}\right) - f_S(x_S)\right], \tag{8}$$

em que N é o conjunto de todas as covariáveis e x_S é o vetor com todos os valores de entrada das covariáveis do conjunto S.

Uma vez que calcular as contribuições exige todos os subconjuntos possíveis de covariáveis, o aumento exponencial do tempo impede que o valor exato dessas contribuições seja calculado. Para contornar esse problema, Štrumbelj & Kononenko (2014) propuseram um procedimento de amostragem via método de Monte-Carlo para estimar os valores Shapley ao perturbar os valores de entrada de uma observação x que se quer explicar.

Dados o modelo não-interpretável f e a matriz de covariáveis $\mathbf{X}$, o algoritmo consiste em:

1. selecione uma observação x de interesse e uma covariável j, $j = 1,\ldots,p$.
2. para cada $m = 1,\ldots,M$ iteração:

2.1 selecione uma permutação aleatória do conjunto de todas as permutações possíveis entre as covariáveis, i.e., $O \in S(\{1,\ldots,p\})$.

2.2 selecione uma observação aleatória z que pertença à matriz de dados $\mathbf{X}$ e gere uma nova amostra, i.e., $z_0 = \left(z_{(1)},\ldots,z_{(j)},\ldots,z_{(p)}\right)$.

2.3 crie uma nova observação com a covariável j ao substituir os valores de x pelos valores de z após a operação j:

$$x_{+j} = \left(x_{(1)},\ldots,x_{(j-1)},x_{(j)},z_{(j+1)},\ldots,z_{(p)}\right).$$

2.4 crie uma nova observação sem a covariável j repetindo o passo (2.2.3), mas fazendo a substituição a partir da operação j:

$$x_{-j} = \left(x_{(1)},\cdots,x_{(j-1)},z_{(j)},z_{(j+1)},\cdots,z_{(p)}\right).$$

2.5 calcule a contribuição marginal de j ao fazer a diferença entre as predições do modelo complexo com e sem a covariável j:

$$\phi_j^m = \hat{f}(x_{+j}) - \hat{f}(x_{-}j).$$

3. calcule a média das M diferenças e encontre o valor Shapley para x:

$$\phi(x) = \tfrac{1}{M}\sum_{m=1}^{M}\phi_j^m.$$

O valor Shapley encontrado em (3) é interpretado como o quanto a inclusão da j-ésima covariável contribuiu, positiva ou negativamente, para a predição de uma observação x em particular em relação à predição média feita para todos os dados.

Na Teoria dos Jogos é assumido que a grande coalizão, isto é, o conjunto composto por todos os jogadores, é formada e os prêmios de cada subcoalizão é somado (totalizando uma grande premiação). No cenário de modelos preditivos, o valor Shapley garante a distribuição justa do valor predito entre as covariváveis ao satisfazer as propriedades Eficiência, Simetria, *Dummy* e Aditividade. A saber.

- Eficiência: se as contribuições da j-ésima covariável são somadas, elas devem ser igual à diferença entre a predição real de x e a predição média dos dados:

$$\sum_{j=1}^{p}\phi_j = \hat{f}(x) - E_{\mathbf{X}}[\hat{f}(\mathbf{X})].$$

- Simetria: se as j-ésima e k-ésima covariáveis contribuem igualmente em todos os subconjuntos possíveis, as contribuições de cada uma devem ser iguais:

$$\phi\left(S\cup\{x_j\}\right) = \phi\left(S\cup\{x_k\}\right), \forall S \subseteq N\backslash\{j,k\} \Rightarrow \phi_j = \phi_k.$$

- *Dummy*: se a j-ésima covariável não contribui para a predição em qualquer subconjunto em que se faz presente, a contribuição deve ser igual a zero:

$$\phi\left(S\cup\{x_j\}\right) = \phi(S), \forall S \subseteq N \Rightarrow \phi_j = 0.$$

- Aditividade: se duas predições calculadas pelos modelos f e g são combinadas, as contribuições distribuídas devem ser igual à soma das contribuições marginais:

$$\phi_j(f+g) = \phi_j(f) + \phi_j(g), \forall j \in N.$$

Além de distribuir justamente as contribuições, satisfazer esses quatros axiomas significa ter uma teoria sólida por trás dos acontecimentos, fazendo com que o valor Shapley seja considerado um dos métodos de interpretabilidade mais completos em termos de explicação. As desvantagens geralmente esbarram no tempo computacional, que é muito grande devido a todas as permutações possíveis de 2^p subconjuntos de covariáveis.

3.2.9 Explicações Aditivas Shapley

Introduzido por Lundberg & Lee (2017), o método Explicações Aditivas Shapley, popularmente conhecido como SHAP (*SHapley Additive exPlanations*, em inglês), é uma abordagem unificada de vários métodos para interpretar a previsão de qualquer observação x como a soma das contribuições individuais de cada covariável de um modelo caixa-preta.

Faz sentido o objetivo ser exatamente igual ao método dos valores Shapley, pois o SHAP usa os valores Shapley (e toda a Teoria dos Jogos) para fazer suas explicações. Aqui, a diferença é que são usados métodos auxiliares que possibilitam flexibizar seu algoritmo para cada tipo de modelo (linear, baseado em árvores, de aprendizagem profunda etc).

O método subjacente ao SHAP que calcula as contribuições das predições independente do modelo é o KernelSHAP, que assume independência entre as covariáveis e linearidade do modelo a fim de simplicar o cálculo dos valores SHAP. De forma geral, KernelSHAP é um método agnóstico que estima os valores SHAP a partir dos conceitos combinados de LIME (seção 3.2.7) e valores Shapley. Nesse cenário, as contribuições são calculadas por meio do ajuste de um modelo substituto local em que a função de perda é dada por

$$L(f,g,\pi_x) = \sum_{z' \in Z} \left[f(h_x(z')) - g(z') \right]^2 \pi_x(z')). \tag{9}$$

Para efeitos didáticos, é válido citar o significado de cada parcela dessa função. O conhecimento de cada uma delas permitirá, adiante, entender com mais clareza como funciona o "SHAP padrão" a partir do momento em que se imputam os dados até as interpretações geradas. Assim, f é o modelo a ser explicado, g é o modelo substituto interpretável, z' é um vetor de zeros ou 1 denominado "coligação", $h_x(z')$ é uma função que mapeia um vetor de coligação para observações de dados correspondentes, Z é o conjunto de dados de treinamento, $\pi(z')$ é o peso atribuído aos vetores de coligação e $g(z')$ é uma combinação linear entre os vetores de coligação e os valores Shapley.

A estratégia adotada aqui é selecionar uma observação x que se quer explicar e criar um vetor de coligação z' de zeros e 1, atribuídos de acordo com o mapeamento do espaço de entrada original: será 1 se a j-ésima covariável estiver presente e 0 se ela estiver ausente. Os K

vetores de coligação z' serão usados como os dados para treinar um modelo de regressão linear, usado como o modelo substituto local, enquanto a variável resposta será dada pela predição das coligações. No entanto, antes de realizar o ajuste do modelo, é necessário ponderar os vetores de coligação com o *kernel* SHAP, que é o que garantirá que os valores gerados sejam compatíveis com Shapley (Lundberg & Lee, 2017). Como resultado, os coeficientes de regressão estimados, ao otimizar a função de perda dada em (9), são os valores SHAP.

Em particular, tem como algoritmo:

1. nos dados de treinamento, selecione uma amostra com k vetores de coligação, i.e.,

$$z'_k \in \{0,1\}^M,\ k \in \{1,\ldots,K\},$$

   em que M é o tamanho da coligação.

2. converta os valores das covariáveis do vetor de coligação para os quais $z' = 1$ pelos valores da amostra original, substituindo os demais por valores de uma observação amostrada aleatoriamente.

3. calcule a média das predições dos vetores de coligação:

4. calcule o peso para cada z'_k por meio do *kernel* SHAP:

$$\pi_x(z') = \frac{M-1}{\binom{M}{|z'|}|z'|(M-|z'|)}.$$

5. ajuste o modelo linear ponderado com os dados selecionados em (1), a predição obtida em (3) e os pesos encontrados em (4) e obtenha as explicações como representações de uma soma linear de contribuição das covariáveis.

$$g(z') = \varphi_0 + \sum_{j=1}^{M} \varphi_j z'_j.$$

Se os valores Shapley encontrados no passo (5) são combinados, obtém-se uma matriz de valores Shapley em que as colunas são as covariáveis e cada linha representa a explicação de uma observação. Analisando essa matriz de maneira integral, a interpretação para o modelo é global.

Para isso, como alternativa restrita para modelos baseados em árvores, Lundberg et al. (2018) propuseram uma implementação eficiente do algoritmo SHAP denominada TreeSHAP, um método capaz de calcular os valores Shapley de forma muito mais rápida e exata. Na prática, o número usado para fazer as permutações não é da forma exponencial, mas depende exclusivamente da profundidade das árvores do modelo que está sendo avaliado. Com TreeSHAP, é possível implementar ferramentas gráficas muito poderosas e de fácil intuição que auxiliam na tarefa de explicar. Vejamo-as a seguir.

3.2.9.1 Gráfico SHAP da importância da covariável

A ideia deste gráfico é bem simples, bastando apenas somar os valores Shapley absolutos de cada i-ésima observação, isto é

$$I_j = \sum_{i=1}^{n} |\phi_j^(i)|,$$

e ordenar as covariáveis segundo seu grau de importância de forma decrescente. A covariável com o maior valor Shapley absoluto é a mais importante.

O gráfico SHAP da importância da covariável é uma alternativa ao método da importância da covariável por permutação (seção 3.2.5). Enquanto no segundo cenário a importância da covariável é medida pela variação na taxa de erro de predição ao permutá-la, o SHAP atribui as importâncias de acordo com o total absoluto dos valores Shapley.

3.2.9.2 Gráfico SHAP de resumo

No gráfico de resumo são combinadas as informações de importância da covariável e qual seu efeito sobre a predição a depender de seu valor. No eixo-x estão os valores Shapley e cada ponto plotado é um valor Shapley calculado para uma observação e covariável, sendo a cor determinada pela magnitude, se alta ou baixa, do valor correspondente àquela mesma observação e covariável. Além disso, os conjuntos de pontos são perturbados a fim de se ter indícios da distribuição dos valores Shapley em cada covariável, dispostas no gráfico de forma decrescente, como no gráfico SHAP da importância da covariável.

3.2.9.3 Gráfico SHAP de dependência

Neste gráfico, os seguintes pontos são plotados:

$$\left\{ \left(x_j^{(i)}, \phi_j^{(i)} \right) \right\}_{i=1}^{n},$$

em que x_j é o valor da j-ésima covariável e ϕ_j é o valor Shapley correspondente da i-ésima observação.

Alternativamente aos gráficos de dependência parcial (seção 3.2.1) e de efeitos locais acumulados (3.2.3), o gráfico SHAP de dependência apresenta os efeitos médios de uma covariável e sua variabilidade em relação ao valores Shapley. No caso em que seja feita a interação entre duas covariáveis, i e j, a variância no eixo-y é ainda mais destacada.

Para calcular o efeito dessa interação, são somados os efeitos individuais de todas as covariáveis, exceto i e j, e subtraídos os efeitos próprios de i e j. Nessa situação, o valor Shapley para a interação entre covariáveis é dado por

$$\phi_{i,j} = \sum_{S \subseteq N \setminus \{i,j\}} \frac{|S|!(|N| - |S| - 2)}{2(N-1)!} \delta_{i,j}(S),$$

em que $\delta_{i,j}(S) = f_x(S \cup \{i,j\}) + f_x(S) - f_x(S \cup \{i\}) - f_x(S \cup \{j\})$.

3.2.9.4 Clusterização de valores SHAP

Fazer clusterização usando valores Shapley significa agrupar pontos de dados por semelhança em explicação, neutralizando o efeito das diferentes escalas das covariáveis. Por padrão, o algoritmo da clusterização de valores SHAP usa o método de clusterização hierárquica acumulativa.

Assim como os valores Shapley, o SHAP também satisfaz "propriedades de justiça" bem semelhantes àquelas vistas na seção anterior. Neste caso, difere-se ao levar em consideração os vetores de coligação.

- Precisão local: o modelo substituto local $g(x')$ corresponde ao modelo a ser explicado $f(x)$ quando $x = h_x(x')$, com $\phi_0 = f(h_x(0))$ sendo a saída do modelo em que todas as entradas são ausentes, i.e., iguais a zero:

$$f(x) = g(x') = \phi_0 + \sum_{j=1}^{M} \phi_j x'_j.$$

- Falta: se a j-ésima covariável está presente nas entradas simplificadas, a falta impõe covariáveis ausentes na entrada original para não atribuir impacto incorretamente.

$$x'_j = 0 \Rightarrow \phi_j = 0.$$

- Consistência: seja $f_x(z') = f(h_x(z'))$ e $z' \backslash j$ denotando $z'_j = 0$. Para quaisquer dois modelos f e f', se a contribuição marginal da j-ésima covariável aumentar ou se manter ao alterar o modelo, o valor de Shapley não deve diminuir:

$$f'_x(z') - f'_x(z' \backslash j) \geq f_x(z') - f_x(z' \backslash j), \forall z' \in \{0,1\}^M \Rightarrow \phi_j(f', x) \geq \phi_j(f, x).$$

Embora o SHAP carregue as mesmas vantagens dos valores Shapley, ele inova ao propôr o TreeSHAP como uma implementação rápida e eficiente para os modelos baseados em árvores. Assim, torna-se viável calcular os valores Shapley mesmo em um banco de dados que tenham muitas observações e covariáveis. Por outro lado, o "SHAP padrão" (KernelSHAP) é muito lento, impossibilitando a aplicação dos métodos de interpretação global. Além disso, ao desconsiderar possíveis correlações entre covariáveis, há o risco em se fazer ponderações significativas em pontos de dados inverossímeis.

4 Resultados

5 Discussão e Conclusões

Referências

Alvarez-Melis, D. & Jaakkola, T. S. (2018). On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049*.

Amorim, W. N. d. (2019). *Ciência de dados, poluição do ar e saúde*. PhD thesis, Universidade de São Paulo.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Doshi-Velez, F. & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Fisher, A., Rudin, C., & Dominici, F. (2018). Model class reliance: Variable importance measures for any machine learning model class, from the" rashomon" perspective. *arXiv preprint arXiv:1801.01489*, 68.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*.

Friedman, J. H., Popescu, B. E., et al. (2008). Predictive learning via rule ensembles. *Annals of Applied Statistics*, 2(3):916–954.

Izbicki, Rafael.; Santos, T. M. d. (2020). *Aprendizado de máquina: uma abordagem estatística*. Câmara Brasileira do Livro, SP, Brasil, 2ª edition.

James, Gareth; Witten, D. H. T. & Tibshirani, R. (2013). *An Introduction to Statistical Learning with Applications in R*. Springer.

Lundberg, S. M., Erion, G. G., & Lee, S.-I. (2018). Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*.

Lundberg, S. M. & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pages 4768–4777.

Molnar, C. (2019). *Interpretable Machine Learning - A Guide for Making Black Box Models Explainable*.

Morettin, Pedro A.; Singer, J. M. (2020). *Introdução à Ciência de Dados - Fundamentos e Aplicações*.

Obermeyer, Z., Powers, B., Vogeli, C., & Mullainathan, S. (2019). Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "why should i trust you?"explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Shapley, L. S. (2016). *17. A value for n-person games*. Princeton University Press.

Štrumbelj, E. & Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665.