PREDIÇÃO DO RISCO DE REPROVAÇÃO NO ENSINO SUPERIOR USANDO ALGORITMOS DE *MACHINE LEARNING*

Andréa Ferreira da Silva[1] Aléssio Tony Cavalcanti de Almeida[2] Hilton Martins de Brito Ramalho[3]

RESUMO
Esta pesquisa propõe identificar o risco de reprovação de discentes do ensino superior usando algoritmos de Machine Learning (ML). Com base nos registros administrativos e acadêmicos da Universidade Federal da Paraíba (UFPB) e da Plataforma Lattes, para o período de 2010 a 2016 da disciplina de cálculo diferencial e integral I, foi verificado que os modelos com a melhor performance de previsão foram *Ridge*, Regressão Logística, *LASSO* e *Elastic Net*, sem diferenças estatísticas de desempenho entre si. A partir da modelagem sobre os dados de treinamento, os resultados encontrados explicitam que, das 1.903 observações que compõem um novo conjunto de dados, o conjunto de teste, a frequência dos alunos com status (reprovados e aprovados) previstos corretamente pela *Accuracy* foi de 67%, em ambos os modelos. Por sua vez, 65% dos discentes foram previstos corretamente como reprovados (*Sensitivity*). Esses achados ratificam que os algoritmos de ML podem ser instrumentos viáveis para auxiliar ações pedagógicas e gerenciais acadêmicas preventivas que visem a redução dos índices de reprovações no ensino superior.
Palavras-chave: Predição de Risco. Reprovação. Ensino Superior. *Machine Learning*.

ABSTRACT
This research proposes to identify the risk of failing higher education students using Machine Learning (ML) algorithms. Based on the administrative records of the Universidade Federal da Paraíba (UFPB) and Plataforma Lattes, for the period 2010-2016 of the discipline of differential and integral calculus I, it was verified that the models with the best forecasting performance were Ridge, Logistic Regression, LASSO and Elastic Net, with no statistical differences in performance between them. From the modeling on the training data, the results found explain that, of the 1,903 observations that make up a new data set, the test set, the frequency of students with status (failed and approved) correctly predicted by Accuracy was 69 %, on both models. In turn, 72% of students were correctly predicted as failing (Sensitivity). These findings confirm that ML algorithms can be viable instruments to assist preventive academic management and pedagogical actions aimed at reducing failure rates in higher education.
Keywords: Risk Prediction. Disapproval. University education. Machine Learning.

Área 8: Microeconomia, Métodos Quantitativos e Finanças.
JEL classification: C52, D04, I23.

1 INTRODUÇÃO

A retenção e a evasão escolar são uma realidade para muitos dos discentes nos cursos de graduações das universidades públicas no Brasil, consideradas problemas complexos nas Instituições de Ensino Superior (IES), o que aumenta os seus custos de provisão. Várias pesquisas mostram que esses problemas são universais e que devem envolver, para a sua solução, diferentes níveis de intervenção, desde aquelas em nível da família e do indivíduo até as relacionadas com os insumos escolares e diretrizes da educação (Gomes-Neto & Hanushek, 1994; Leon & Menezes-Filho, 2002; Sampaio, Sampaio, Mello & Melo 2011; Diogo et al., 2016).

As taxas de reprovação são, para muitos autores, um dos maiores problemas no sistema de ensino brasileiro (Júnior; Faria & Lima, 2012). De acordo com Souza, Ponczek, Oliva e Tavares (2012), o discente,

[1] Doutora em Economia Aplicada (PPGE-UFPB). Bolsista de Pós-Doutorado no Instituto de Saúde Coletiva (ISC-UFBA) Professora temporária do Departamento de Economia da Universidade Regional do Cariri (URCA-UDI).
[2] Doutor em Economia Aplicada. Professor do Programa de Pós-Graduação em Economia, UFPB.
[3] Doutor em Economia Aplicada. Professor do Programa de Pós-Graduação em Economia, UFPB.

ao evadir de um determinado curso, pode ingressar em um novo logo em seguida. Por outro lado, a deficiência no aprendizado poderá acompanhá-lo e acarretar novas reprovações, aumentando a probabilidade de evasão.

Nos últimos anos, no Brasil, é possível observar uma crescente oferta de cursos. De acordo com os dados relativos ao Censo da Educação Superior, referente ao ano de 2015, divulgados pelo Instituto Nacional de Estudos e Pesquisa Anísio Teixeira (Inep), 2.368 IES[4] ofertaram 8 milhões de vagas, correspondentes a 33 mil cursos de graduação, o que representa um incremento de 11% em relação às vagas ofertadas em 2010[5]. Muito embora tenha se observado o referido aumento, boa parte dos alunos ingressantes não chegam a concluir os cursos. Segundo dados do Censo da Educação Superior de 2017, a taxa de conclusão nas instituições públicas gira em torno de 37% e de 36% nas privadas (Inep, 2018).

A evasão é caracterizada como uma interrupção no ciclo de estudos de um estudante, a qual pode gerar prejuízos significativos em vários âmbitos, tais como o acadêmico, o social e o econômico. No que tange ao setor público, ao evadir dos seus cursos, os discentes geram perdas diretas e indiretas, aumentando os custos da diplomação e reduzindo a taxa de retorno dos investimentos em educação. Já no setor privado, a evasão provoca perdas sociais importantes no que tange à redução do estoque de profissionais formados que poderiam aumentar a quantidade de mão de obra qualificada no mercado e, assim, a produtividade da economia (Silva-Filho, Motejunas, Hipólito & Lobo, 2007).

Shirasu e Albuquerque (2015), em estudo sobre os determinantes da evasão e da repetência, constataram que a falta de interesse em estudar e as repetidas reprovações são os fatores com maiores influências nas taxas de evasão discente. Para Gomes-Neto e Hanushek (1994), a reprovação, ocasionando repetências, e a evasão estão interligadas, de maneira que a combinação entre elas tem sido identificada como uma das falhas do sistema educacional no Brasil, pois ratifica a ineficiência dos gastos públicos neste sistema. Desse modo, o problema da reprovação dos estudantes pode ser, por muitas vezes, um dos fatores determinantes da evasão.

Dada a abrangência de fatores, do ponto de vista das IES, prever a reprovação acadêmica é necessário, pois esta ação pode proporcionar medidas preventivas com o propósito de evitar novos resultados negativos no fim de mais um semestre. Especificamente, é um indicador que não só mede a aprendizagem e o desempenho do aluno, mas também o seu impacto em outros desequilíbrios. Mas como prever de forma precisa o risco de reprovação? Quais são os principais fatores que fazem com que os estudantes sejam aprovados ou não? Como adotar medidas preventivas para que essa tendência seja reduzida? Qual é o papel dos gestores educacionais?

A identificação de discentes com maior risco de sofrer reprovação nas disciplinas com os maiores índices de retenção pode estar diretamente relacionada à necessidade de intervenções na educação no ensino superior, buscando reduzir não apenas a retenção, mas, consequentemente, a evasão, para que, assim, possa evitar custos ocasionados em decorrência desse comportamento nas universidades. Nesse contexto, a análise preditiva pode auxiliar a ponderação entre benefícios e danos, com a finalidade de auxiliar administradores educacionais na formulação de políticas públicas direcionadas às intervenções preventivas, como, por exemplo, reforço escolar, acompanhamento pedagógico, curso de férias, entre outros.

Dessa maneira, um dos grandes desafios é desenvolver uma estratégia eficiente, passível de operacionalização prática, que consiga prever o resultado dos discentes, de modo a permitir uma intervenção prévia de professores, coordenadores e outros responsáveis institucionais com o escopo de evitar ou minimizar problemas futuros de reprovação e de possível evasão. Uma vez que as IES armazenam dados acadêmicos e socioeconômicos dos seus estudantes, é possível realizar diversas análises na busca por padrões e características relacionadas com a condição de reprovação do alunado. Por sua vez, como o processo exige uma investigação baseada na extração de conhecimento em um extenso volume de dados, as diferentes técnicas de aprendizagem de máquina, *Machine Learning* (ML), apresentam-se como opções viáveis para realizar essa tarefa.

Assim, esta pesquisa tem como objetivo classificar, de maneira precoce, os discentes com risco de

[4] Centros Universitários, Faculdades, Universidades, Institutos Federais (IFs) e Centros Federais de Educação Tecnológica (Cefets).

[5] Segundo informações do Censo da Educação Superior, em 2010, 29.507 cursos de graduação ofertaram 6.379.299 vagas (Inep, 2018).

reprovação a partir da aplicação dos algoritmos de *ML*. Para tanto, foram utilizados registros administrativos e acadêmicos da Universidade Federal da Paraíba (UFPB), de 14 semestres letivos (2010 a 2016) e de mais de 6.340 matrículas em uma disciplina de alta retenção na mencionada instituição: Cálculo Diferencial e Integral I.

Na UFPB, diversas graduações possuem em suas grades curriculares disciplinas da área de matemática, sendo para muitas uma base essencial para a formação do aluno (como nos cursos de Matemática, Física, Ciência da Computação, Economia e Engenharias). Devido à sua relevância, cabe prever os possíveis riscos que o fracasso ou insucesso dos discentes nesta disciplina possa vir a contribuir no processo de evasão nesta etapa de ensino[6].

Na literatura de economia da educação, muito se tem debatido acerca da evasão no ensino superior e suas principais causas, principalmente no âmbito da inferência causal (Leon & Menezes-Filho, 2002; Mello & Melo, 2011; Diogo et al., 2016). Contudo, são escassos estudos do ponto de vista preditivo[7]. Costumeiramente, os trabalhos que analisam a evasão adotam uma abordagem estatística mais tradicional, como, por exemplo, Sampaio et al. (2011), preocupando-se com os fatores determinantes da variável de desfecho. Dessa maneira, pensando em contribuir para a redução dos problemas de reprovação e, consequentemente, de evasão, este estudo adota os algoritmos de ML (*LASSO*, k-vizinhos mais próximos, classificação *Naïve Bayes*, árvore de decisão, entre outros) para fazer uma classificação preditiva mais precisa dos indivíduos com potencial problema de rendimento acadêmico. Por fim, adotou-se um teste de hipótese para analisar a significância estatística das divergências de resultados dos classificadores, o teste de McNemar.

Após esta seção introdutória, este ensaio contempla mais quatro Seções. A Seção 2 apresenta as variáveis que compõem a base de dados e suas descrições. A Seção 3 descreve as estratégias empíricas que foram aplicadas referentes aos algoritmos abordados do aprendizado de máquina e os critérios de seleção do modelo. A Seção 4, de resultados, contém as subseções referentes à comparação, seleção e avaliação do modelo com melhor previsão, e por fim, a Seção 5 trata das conclusões do estudo.

2 BASE DE DADOS E DESCRIÇÃO DAS VARIÁVEIS

As informações usadas nesta pesquisa fazem parte dos microdados oriundos da Superintendência de Tecnologia da Informação (STI) da Universidade Federal da Paraíba (UFPB) e contêm características sobre os discentes que ingressaram nos cursos de graduação e que demandaram a disciplina de Cálculo Diferencial e Integral I por semestre, no período de 2010 a 2016, como também suas notas do vestibular e características dos respectivos docentes. Logo, a base não está dividida por cursos, e sim por disciplina. Destaca-se que os estudantes tiveram sua identificação preservada.

A base de dados compõe informações de 3.109 discentes (49%) que foram reprovados na disciplina de Cálculo Diferencial e Integral I e de 3.233 (51%) alunos que foram aprovados no mesmo período, sendo composta, assim, por um total de 6.342 observações. Com o propósito de evitar um possível viés nos modelos de previsões, a amostra está restrita a alunos que cursaram apenas uma única vez a disciplina supracitada. A primeira classe compõe os estudantes que não atingiram o nível suficiente de rendimento, isto é, não obtiveram a média final igual 7 na disciplina no fim do período letivo ou foram reprovados por falta. Por sua vez, a segunda classe é composta pelos discentes que foram aprovados com média final igual ou superior a 7.

Logo, para alcançar o objetivo proposto da pesquisa, a variável de resposta, que foi estimada nos algoritmos de classificação de ML, é uma variável binária e assume 1 quando o discente apresenta o *status* de matrícula "reprovado" em Cálculo Diferencial e Integral I, e 0, caso contrário. O banco de dados adotado para predizer o desempenho dos estudantes é composto pelas variáveis que detêm informações dos discentes da UFPB nos semestres de 2010.1 a 2016.2, como também por outras características em nível dos docentes, da turma, do curso e do centro. A seguir, a Tabela 1 reporta as variáveis que compõem cada uma das dimensões adotadas neste estudo.

[6] Uma vez modelado e, posteriormente, testado e implementado no sistema acadêmico das instituições, o modelo de risco de reprovação realizado para a disciplina de Cálculo I poderia ser expandido para outras disciplinas.

[7] Modelos preditivos têm sido utilizados na literatura, mas aplicados em outras áreas, como finanças, comércio eletrônico e macroeconomia (Kleinberg, Mullainathan & Raghavan, 2016; Björkegren & Grissen, 2018).

Tabela 1 – Descrição das Variáveis.

Dimensão	Variáveis	Descrição	Fonte
Discente	Nota Vestibular	Nota do vestibular total, variando de 0 a 1000.	STI/UFPB
	Nota Vest. Mat.	Nota do vestibular total na prova de matemática, variando de 0 a 1000.	STI/UFPB
	Casado	*Dummy*: casado assume 1, e 0, caso contrário.	STI/UFPB
	Migrante	*Dummy*: migrante assume 1, e 0, caso contrário.	STI/UFPB
	Raça	*Dummy*: branco assume 1, e 0, caso contrário.	STI/UFPB
	Sexo	*Dummy*: feminino assume 1, e 0, caso contrário.	STI/UFPB
	Idade Ingresso	Idade no semestre de ingresso, variando de 15 a 71 anos.	STI/UFPB
	Cotista	*Dummy*: cotista assume 1, e 0, caso contrário.	STI/UFPB
	Período Ingresso	*Dummy*: 2º semestre assume 1, e 0, caso contrário.	STI/UFPB
	Forma de Ingresso	*Dummy*: Enem assume 1, e 0, caso contrário (PSS).	STI/UFPB
Docente	Tempo de Grad.	Calculada a partir do ano de conclusão da primeira graduação varia de 1 a 40 anos.	CNPq
	Doutorado	*Dummy*: doutorado assume 1, e 0, caso contrário.	CNPq
	Publicação no Ano	*Dummy*: publicação no ano assume 1, e 0, caso contrário.	CNPq
	Estrangeiro	*Dummy*: estrangeiro assume 1, e 0, caso contrário.	CNPq
	Dedic. Exclusiva	*Dummy*: dedicação exclusiva assume 1, e 0, caso contrário.	STI/UFPB
	Sexo	*Dummy*: feminino assume 1, e 0, caso contrário.	STI/UFPB
Curso	Local do *Campus*	*Dummy*: João Pessoa assume 1, e 0, caso contrário.	STI/UFPB
	EF do Curso	*Dummies* por curso.	STI/UFPB
	EF do Centro	*Dummies* por centro.	STI/UFPB
Turma	Turno	*Dummy*: noturno assume 1, e 0, caso contrário.	STI/UFPB
	Carga Horária	*Dummy*: 90 hrs assume 1, e 0, caso contrário.	STI/UFPB
	Média N. Vestibular	Nota do vestibular total média, variando de 0 a 1000.	STI/UFPB
	Média N. Vest. Mat.	Nota do vestibular total média na prova de matemática, variando de 0 a 1000.	STI/UFPB
	Taxa de Cotista	Percentual de discentes cotista na turma.	STI/UFPB

Fonte: Elaboração própria a partir dos microdados do STI/UFPB e Plataforma *Lattes* do CNPq.

Como as médias ao final de cada semestre na disciplina são utilizadas como principal e único critério para a reprovação/aprovação do discente, e como o fracasso na disciplina de Cálculo pode estar relacionado também com alguma deficiência agregada no conhecimento geral oriundo do ensino básico, optou-se em selecionar variáveis que apresentam o desempenho no vestibular como um todo e, de maneira mais específica, o desempenho na nota em matemática [(i) *Nota Vestibular* e (ii) *Nota Vestibular Matemática*].

Algumas das variáveis que compõem o conjunto de características dos docentes da UFPB foram construídas a partir das informações disponibilizadas e coletadas na Plataforma *Lattes* do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq)[8]. De modo específico, a variável *Estrangeiro* foi construída baseada no país onde a universidade na qual foi concluída a graduação situa-se. Ao adotar a nacionalidade do docente, busca-se identificar se o professor estrangeiro pode ser um problema na transmissão oral do conhecimento da disciplina, visto que está já é considerada um assunto de difícil assimilação entre os discentes.

A disciplina de Cálculo Diferencial e Integral I compõe a grade curricular de 21 cursos de graduação da UFPB, que fazem parte de seis centros de ensino. Foram construídas 21 *dummies* referentes aos cursos: (1) Ciência da Computação; (2) Ciências Atuariais; (3) Ciências Econômicas; (4) Ciência Ambiental; (5) Engenharia Civil; (6) Engenharia de Alimentos; (7) Engenharia da Computação; (8) Engenharia de Energias Renováveis; (9) Engenharia de Materiais; (10) Engenharia de Produção Mecânica; (11) Engenharia de Produção; (12) Engenharia Elétrica; (13) Engenharia Mecânica; (14) Engenharia Química; (15) Estatística; (16) Física; (17) Matemática (Omitida); (18) Matemática (Licenciatura); (19) Matemática Computacional; (20) Química; e (21) Química Industrial. Como também seis *dummies* dos centros: (1) Centro de Ciências Aplicadas e Educacional (CCAE); (2) Centro de Ciências Exatas e da Natureza (CCEN) (Omitida); (3) Centro de Ciências Sociais e Aplicadas (CCSA); (4) Centro de Energias e Alternativas e Renováveis (CEAR); (5) Centro de Informática (CI); e (6) Centro de Tecnologia (CT).

[8] Ver <http://lattes.cnpq.br/>.

3 ESTRATÉGIA EMPÍRICA – OS ALGORITMOS DE *MACHINE LEARNING*

Os algoritmos de ML vêm ganhando espaço entre as pesquisas dos economistas nos últimos anos (Varian, 2014; Bajari, Nekipelov, Ryan & Yang, 2015a; Bajari, Nekipelov, Ryan & Yang, 2015b; Athey, 2018). Por tratar diretamente de problemas que lidam com *big data*, a aplicação do ML tem se tornado eficiente quando o principal objetivo do pesquisador é a previsão do risco de um evento ocorrer, particularmente no que se refere no grau de automatização do processo de modelagem, estimação, teste e escolha do melhor modelo de predição.

A discussão sobre ML e o seu desempenho preditivo passa pelo *trade-off* entre viés e variância (*bias-variance trade-off*) (Hastie et al., 2009). A relação custo-benefício nesse *trade-off* ocorre quando se torna possível reduzir as incertezas das predições e projeções ao custo de um aumento do viés nos estimadores. Considerada uma das técnicas de *data mining*, o ML avança em relação às abordagens estatísticas mais tradicionais no que tange às melhorias em fazer previsões em conjuntos de dados cada vez maiores e mais complexos, bem como ao enfoque para avaliação e seleção dos modelos.

A análise preditiva, por sua vez, baseia-se na aplicação de algoritmos em estruturas de dados existentes na busca por estimar o risco de eventos futuros ocorrerem com base em experiências passadas, e, assim, gerar tomadas atuais de decisões (Hastie et al., 2009; Kuhn & Johnson, 2013). Contudo, a acurácia dessas estimativas é um dos aspectos mais importantes do modelo. Por isso, uma boa parte do debate na literatura em ML se dedica às métricas utilizadas para reduzir a parcela redutível do erro de previsão dos estimadores, uma vez que a parcela irredutível não pode ser trabalhada, no caso, por exemplo, da omissão de variáveis no modelo.

Os métodos de ML podem ser divididos em aprendizado supervisionado e aprendizado não supervisionado. Neste estudo, foram aplicados métodos de aprendizado supervisionado, os quais reúnem métodos de estimação, em que cada observação do preditor da base de dados mensurado por X_i, $i = 1, 2, ..., n$, há uma variável de interesse (dependente), Y_i. Ou seja, o principal objetivo é ajustar o modelo que relacione os preditores, X, a variável de resposta, Y, com a finalidade de prever o evento em observações futuras. Por outro lado, o aprendizado não supervisionado detém os métodos em que, para cada observação das covariadas, não se tem a variável de resposta correspondente (James *et al.*, 2013; Athey, 2018).

O tipo de variável a ser predita pode ser definido em dois subgrupos diferentes no aprendizado supervisionado: regressão, para variáveis quantitativas; e classificação, para variáveis categóricas ou qualitativas. Em ambos os casos, o ajuste dos modelos de ML pode ser descrito nas seguintes etapas: (i) divisão (aleatória - dependendo do tamanho da base) dos dados em conjuntos de treinamento e realização de teste na etapa de pré-processamento; (ii) na etapa de aprendizado, ocorre a seleção do modelo com melhor previsão em dados de treinamento, ante uma gama de algoritmos; (iii) na terceira etapa, a predição da resposta de interesse na base de teste, e; por fim, (iv) a avaliação do melhor modelo em novos dados (Hastie et al., 2009; James *et al.*, 2013; Raschka, 2017).

Segundo Raschka (2017), a divisão da amostra em conjunto de dados de treinamento e teste é realizada com o intuito de verificar se o modelo apresenta boa predição não apenas nos dados que foram utilizados no ajuste (treinamento), mas também na capacidade de generalização para uma nova amostra (teste). Em geral, dependendo do tamanho da base de dados, as divisões mais adotadas seguem os seguintes padrões: 60:40; 70:30 ou 80:20. Logo, quanto maior o número de observações, maior será o conjunto de dados utilizado na etapa de treinamento.

Como o objetivo deste estudo baseia-se na ideia de prever o nível de reprovação dos discentes, o foco será um problema de classificação, de forma que foi feita uma previsão acurada da variável de interesse, nesse caso: reprovação na disciplina, denotada por $\hat{Y}$, a partir dos valores associados ao vetor de preditores, $\boldsymbol{X}$, contendo informações sobre o aluno, docentes, turma, curso e centro. O problema de classificação baseia-se na divisão do espaço amostral dos preditores em grupos relacionados às categorias de resposta de interesse. A fronteira que define a divisão entre esses grupos é denominada de classificador, o qual representa o algoritmo que estima o modelo preditivo (Hastie et al., 2009).

O aprendizado de modelos preditivos é composto por dois principais objetivos: selecionar e avaliar. No que se refere a selecionar, a performance de diferentes modelos é avaliada por meio de critérios de medidas de desempenho para que, a partir de um equilíbrio entre viés-variância, seja selecionado o modelo

que resulta em uma melhor acurácia e desempenho no conjunto de treinamento. Já no que se refere ao objetivo de avaliar, após a definição da melhor performance, busca-se estimar o modelo em novas observações, na base de teste (Hastie et al., 2009).

Na literatura de ML há um consenso de que não existe um algoritmo que seja capaz de ter uma boa performance em todas as aplicações. Logo, é importante conhecer e comparar os diversos métodos com características diferentes entre si para selecionar o modelo com a melhor performance preditiva para o problema abordado (Lantz, 2013; Kuhn & Johnson, 2013; Izbicki & Santos, 2018). De um modo geral, na etapa de aprendizado, os algoritmos de ML podem ser divididos nas seguintes categorias: lineares (regressão logística); não lineares (*K – nearest neighbors*, *naïve bayes classifier*, *neural network* e *support vector machines*); e modelos baseados em árvores de decisão (*regression trees*, *classification trees*, *bagging*, *random forest* e *gradiente boosting*).

3.2 *Penalized methods*

A literatura de *ML* afirma que, muito embora o modelo de regressão linear seja um bom ponto de partida para a compreensão das abordagens de ML, uma vez que muitos dos algoritmos mais sofisticados podem ser vistos como generalizações ou extensões, há algumas características indesejáveis desse método (Han, Kamber & Mining, 2001; Hastie et al., 2009; Kuhn & Johnson, 2013; Lantz, 2013; Izbicki & Santos, 2018). Hastie, Tibshirani e Friedman (2009) destacam algumas delas: (i) a existência de um elevado número de preditores pode ter como consequência preditores autocorrelacionados ou, ainda, problemas relacionados ao número de preditores ser maior do que o de observações e, a partir daí, o modelo terá infinitas soluções e variância tendendo ao infinito; (ii) embora o método MQO tenha como característica estimadores não viesados, pode apresentar também elevada variância, comprometendo a acurácia das previsões e a interpretação do modelo.

Ante ao contexto da elevada variância que pode ocorrer ao aplicar o método de regressão linear para previsões, e como o método de seleção do subconjunto de preditores não sana o referido problema, Hastie et al. (2009) indicam que os métodos de *Shrinkage* são os mais adequados para abordar esse problema, uma vez que são mais contínuos e não sofrem tanto com a elevada variância. Este conjunto de regressores faz parte da classe de estimadores dos métodos com penalidades, os *penalized methods*.

Desse modo, a solução busca penalizar os coeficientes estimados a fim de limitar a variância ao custo de um aumento insignificante de viés (James *et al.*, 2013). O ponto central tratado baseia-se no *trade-off* entre viés e variância, em que é possível obter métodos com menor variância ao acrescentar viés aos estimadores. De acordo com Kuhn e Johnson (2013), através de um pequeno viés nos preditores, torna-se possível reduzir a variância do modelo e, assim, obter uma melhora na performance de previsão em novas observações. Os métodos mais conhecidos de penalização são *ridge*, *LASSO* e *elastic net*. Dependendo do tipo de pena, alguns coeficientes podem ser estimados como exatamente zero.

A regressão *ridge* reduz os coeficientes da regressão impondo uma penalidade ao seu tamanho. De outra maneira, as estimativas dos parâmetros são oriundas da minimização da função perda (SQR) com penalização quadrática (James *et al.*, 2013). É muito semelhante ao MQO, exceto que os parâmetros são estimados minimizando uma quantidade ligeiramente diferente. O algoritmo de *ridge* estimará $\hat{\beta}^{ridge}$ minimizando:

$$\hat{\beta}^{ridge} = \underset{\beta}{argmin}\left\{\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2\right\} \quad (1)$$

Para $\lambda \geq 0$, λ é um parâmetro de *tunning*, ou ajuste, que controla a quantidade de *shrinkage*, ou seja, quanto maior o valor de λ, maior a penalização ou tamanho do encolhimento. Os coeficientes são encolhidos em direção a zero. O intercepto β_0 assumirá o valor médio da resposta de interesse quando todas as covariadas assumirem valor zero, pois este não sofre a penalização, já que o objetivo é penalizar apenas os parâmetros associados aos preditores.

O LASSO é um método de *shrinkage* como o *ridge*, com diferenças sutis, mas importantes. Assim como na regressão de *ridge*, pode ser feita a reparametrização da constante β_0 padronizando os preditores, e a solução para $\hat{\beta}_0$ é $\bar{y}$, onde a penalização passa a ser baseada agora no valor absoluto dos parâmetros. A estimativa do *LASSO*:

$$\hat{\beta}^{lasso} = \underset{\beta}{argmin}\{\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \; + \; \lambda \sum_{j=1}^{p}|\beta_j|\} \tag{2}$$

Observe a semelhança com o problema de regressão *ridge* da Equação 1. A penalidade L1 da *ridge* $\sum_{j=1}^{p}\beta_j^2$ é substituída pela penalidade L2, $\sum_{j=1}^{p}|\beta_j|$, do LASSO. Assim, o método *LASSO* também reduz as estimativas dos coeficientes para zero. No entanto, a penalidade em *LASSO* tem o efeito de forçar que algumas das estimativas sejam exatamente zero, podendo excluir alguma variável (James *et al.*, 2013). Este tipo de penalização torna as soluções não lineares em y_i e não apresenta solução analítica para β, ou seja, não há como expressar, de forma fechada, o vetor dos parâmetros estimados, como ocorre na regressão *ridge* (Hastie et al., 2009).

Por fim, o *elastic net* é um algoritmo que envolve as normas de penalidades L1 e L2 da *shrinkage*, ou seja, é um método de regressão que adota como restrição a combinação linear entre a restrição L1 da *ridge* e a L2 da *LASSO*. O método contribui tanto para a estimação de soluções esparsas quanto para a restrição das estimativas dos parâmetros (KUHN; JOHNSON, 2013).

$$\hat{\beta}^{Enet} = \underset{\beta}{argmin}\{\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \; + \; \lambda_1 \sum_{j=1}^{p}\beta_j^2 + \; \lambda_2 \sum_{j=1}^{p}|\beta_j|\} \tag{3}$$

Kuhn e Johnson (2013) ainda destacam que a validação cruzada *k-fold* pode ser adotada no processo de otimização dos parâmetros λ, nos três casos.

3.3 Regressão Logística

A regressão logística é um dos modelos de probabilidade linear direcionado para prever respostas qualitativas, ou seja, é um método linear para classificação. Deve-se ter um bom desempenho não apenas nos dados de treinamento, mas também em observações de teste que não foram usadas para treinar o classificador. Considere um conjunto de dados padrão (*default*), em que o *default* de resposta cai em uma das duas categorias: sim ou não. Em vez de modelar a variável de resposta Y diretamente, a regressão logística modela a probabilidade de que Y pertence a uma categoria específica, ou seja, a regressão logística modela a probabilidade de *default*.

De acordo com Kuhn e Johnson (2013), para que essa probabilidade seja estimada, a variável de resposta da base de dados de treinamento é modelada a partir de uma distribuição binomial, que tem como parâmetro (p) a probabilidade da ocorrência de uma categoria específica. É de suma importância ressaltar que a probabilidade estimada deve estar no limiar do intervalo [0, 1] e, ao mesmo tempo, apresentar uma relação direta com as covariadas (preditores) para cada observação da base de dados (James *et al.*, 2013; Kuhn & Johnson, 2013). Dessa maneira, para a estimação, utiliza-se a função logística:

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}} \tag{4}$$

a qual é responsável por modelar a relação entre o conjunto de preditores, **X**, e a probabilidade de uma determinada resposta, $p(\boldsymbol{X}) = Pr(Y = k \mid \boldsymbol{X} = x)$. É válido destacar que, no processo de predição, não é necessário que o modelo atenda aos pressupostos. Para Izbicki e Santos (2018), quando o propósito é a inferência, apenas se espera que um bom classificador seja estimado. Embora possam ser aplicados mínimos quadrados (não lineares) para ajustar o modelo logístico da Equação 4, o método da máxima verossimilhança é o mais comum para estimar o vetor dos parâmetros, $\boldsymbol{\beta}$, uma vez que possui as melhores propriedades estatísticas (James *et al.*, 2013).

Por fim, há uma relação entre a fronteira de decisão linear no método de regressão logística e a escolha de um ponto de corte para *p(X)*. Em um exemplo apresentado por James et al. (2013) e destacado por Santos (2018), o ponto de corte determinado para *p(X)* é 0,5, ou seja, indivíduos com *p(X) > 0,5* vão ser classificados como um grupo que apresenta a presença de uma resposta específica, assim como aqueles com *p(X) < 0,5* representam a ausência de determinada resposta. Portanto, a determinação de um ponto de corte irá definir uma fronteira de decisão linear para o modelo de regressão logística e, assim como na

regressão linear, as penalidades *ridge*, *LASSO* e *elastic net* podem também ser aplicadas em conjunto com a regressão logística.

3.4 *K-Nearest Neighbors* (*KNN*)

Considerado um dos algoritmos mais populares do *Machine Learning* (Benedetti, 1977; Stone, 1977), o método *K-Nearest Neighbors* (*KNN*), ou o método dos *K*-vizinhos mais próximos, é uma alternativa não paramétrica tanto para os problemas de classificação como para os de regressão, quando a relação entre a resposta de interesse e os preditores demanda um modelo mais flexível, como, por exemplo, em casos não lineares (Hastie et al., 2009).

A abordagem KNN identifica os KNNs da amostra no espaço preditivo, e a resposta prevista para a nova amostra é, então, a média das respostas dos *K* vizinhos mais próximos (Kuhn & Johnson, 2013). Raschka (2017) destaca que esse método possui duas características importantes que precedem a previsão da resposta de interesse para o novo grupo de observações: a determinação do número de *K* vizinhos mais próximos que serão a vizinhança da nova observação; e a determinação da medida de distância que identificará as *K* observações da base de dados de treinamento mais próximas à nova observação. A definição do número de *K* vizinhos mais próximos é tida como um parâmetro que estabelece o quão bem o ajuste do modelo será generalizado para dados futuros.

No que se refere à determinação da medida de distância, o método KNN básico depende de como o pesquisador define a distância entre as amostras. A distância euclidiana, ou seja, a distância em linha reta entre duas amostras, é a métrica mais comumente utilizada (Kuhn & Johnson, 2013). Assim, após determinar o número de vizinhos e a medida de distância a serem adotados, no método KNN, em problemas de regressão, a resposta é a média das respostas observadas da sua vizinhança, $N_k(x)$, definidas a partir de *K* bases de treinamento *T* com o vetor de covariadas, x_i, dos seus *K* vizinhos mais próximos (Hastie et al, 2009; Izbicki & Santos, 2018). Especificamente, o KNN busca estimar a distribuição de *Y* dado *x*, logo, $\hat{Y}$ é definido da seguinte forma:

$$\hat{Y}(x) = \frac{1}{k}\sum_{x_i \in N_k(x)} y_i \tag{5}$$

em que $N_k(x)$ é o total de vizinhos de *x* definido pelos *K* pontos mais próximos, x_i, do conjunto de treinamento. Por sua vez, nos problemas de classificação, a resposta a ser predita deve ser representada pela classe mais comum observada na vizinhança de x^*. Ou seja, para cada classe da resposta de interesse *j*, calcula-se a probabilidade condicional da nova observação pertencer a *j-ésima* classe através da fração de pontos em $N_k(x)$ cujo valor da resposta é *j*:

$$P(Y = j \mid X = x^*) = \frac{1}{K}\sum_{x_i \in N_k(x)} I(y_i = j) \tag{6}$$

onde a classe predita *j* para $x*$ será representada pela classe que evidenciar a maior probabilidade condicional (Hastie et al, 2009; Santos, 2018).

3.5 *Naïve Bayes Classifier*

Os classificadores *bayesianos*, baseados no teorema de *Bayes*, são classificadores estatísticos que buscam prever a probabilidade de participação na classe, ou seja, a probabilidade de uma determinada "resposta", ou observação, pertencer a uma determinada classe. Ao mesmo tempo, exibiram alta precisão e velocidade quando aplicados a grandes bancos de dados (Han, Kamber & Mining, 2001).

Han, Pei e Kamber (2011) e Lantz (2013) afirmam que há um classificador bayesiano simples conhecido como *naïve bayes classifier*, ou classificador *bayesiano* "ingênuo", comparável em desempenho e performance aos algoritmos dos modelos baseados em árvore de decisão e classificadores de *neural network*. Embora não seja o único método de aprendizado de máquina que utiliza métodos *bayesianos*, é o mais comum.

Os classificadores *naïve bayes* assumem que o efeito de uma característica em uma determinada classe é independente dos valores das outras características. Essa suposição é chamada de *class conditional independence*, independência condicional de classe, e é imposta para simplificar as

análises envolvidas e, por esse motivo, são considerados classificadores "ingênuos".

Inicialmente, o teorema de *bayes* é útil na medida em que fornece uma maneira de calcular a probabilidade posterior, $P(y/\mathbf{X})$, de $P(y)$, $P(\mathbf{X}/y)$ e $P(\mathbf{X})$. Assim, assumindo que **X** seja um vetor de *p* covariadas ou parâmetros e *y* a variável de classe, tem-se o teorema de *bayes* dado por:

$$P(y|\boldsymbol{X}) = \frac{P(\mathbf{X}/y)\ \mathrm{P(y)}}{P(\boldsymbol{X})} \tag{7}$$

Tendo esta teoria como base, de acordo com Han et al. (2011), e supondo que existam *m* classes, *C*1, *C*2, ..., *Cm*, o classificador *naïve bayes*, ou classificador *bayesiano* simples, prevê que o rótulo de classe da tupla **X** é a classe *Ci* se e somente se:

$$P(\boldsymbol{X}|C_i)P(C_i) > P\big(\boldsymbol{X}\big|C_j\big)P\big(C_j\big) \text{ para } 1 \leq j \leq m, j \neq i \tag{8}$$

Em outras palavras, o rótulo de classe previsto é a classe *Ci*, para a qual $P(\mathbf{X}|Ci)$ é o máximo.

3.6 *Support Vector Machines* (*SVM*)

Proposto inicialmente por Cortes e Vapnik (1995) para problemas de classificação e posteriormente por Smola *et al.* (1996) e Drucker *et al.* (1997) para problemas de regressão, o *Support Vector Machines* (*SVM*) é um algoritmo que modela fronteiras não lineares. Considerado uma generalização do método *Maximal Margin Classifier*, classificador de margem máxima, o SVM difere dos demais algoritmos do *Machine Learning*, direcionados para problemas de classificação, por não estimar probabilidades diretamente, e sim classes da resposta de interesse estimadas em novas observações.

A solução para o problema do *Support Vector Classifier* envolve apenas os produtos internos das observações (em oposição às próprias observações). O produto interno de dois *r*-vetores *a* e *b* é definido como $\langle a, b\rangle = \sum_{i=1}^{r} a_i b_i$ (James *et al.*, 2013). Nessa perspectiva, o SVM é considerado um classificador que representa uma extensão do *Support Vector Classifier*, o qual resulta da ampliação do espaço de recurso de uma maneira específica, utilizando *Polynomial Kernel* (Hastie et al., 2009; JAMES *et al.*, 2013).

Assim, quando o *Support Vector Classifier* é combinado com um kernel não linear, ou o *Polynomial Kernel*, o classificador resultante é conhecido como *Support Vector Machines*, o (*SVM*). Hastie et al. (2009) apresentam que o algoritmo SVM pode ser adaptado para problemas de regressão, ou seja, problemas com uma resposta quantitativa, de forma a herdar algumas das propriedades do SVM para problemas de classificação. No que se refere a este último, a função pode ser representada por:

$$f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle x, x_i\rangle \tag{9}$$

Nesse caso, apenas o subconjunto estimado para a_i, referente aos vetores de suporte, será diferente de zero. Ao invés de produtos internos das observações propriamente ditas, a solução depende do produto entre os vetores de mensurações dos preditores. Portanto, diferentes *Kernels*, *polinomial* e *radial*, podem ser adotados para os produtos internos (Hastie et al. 2009).

3.7 *Decision Tree-Based Methods*

Decision Tree-Based Methods, ou métodos baseados em árvore de decisão, consistem em um conjunto de regras utilizadas para estratificar ou segmentar o espaço do preditor em um número simples de regiões, resumindo em uma árvore (Hastie et al., 2009; James *et al.*, 2013). Hastie et al. (2009) afirmam ainda que métodos baseados em árvores de decisão representam boa alternativa de modelo preditivo quando a relação entre as covariadas e a resposta de interesse é complexa e não linear.

A árvore de decisão pode ser aplicada tanto para respostas contínuas, em casos de problemas de regressão, como para respostas categóricas, nos problemas de classificação. Em ambos os casos, para fazer uma previsão para uma dada observação, normalmente se usa a média ou a moda das observações de treinamento na região à qual ela pertence. São simples e úteis para interpretações,

contudo, não são competitivas com as melhores abordagens de aprendizado supervisionadas, uma vez que apresentam um poder preditivo muito baixo quando comparadas aos demais algoritmos (James *et al.*, 2013; Izbicki & Santos, 2018).

3.7.1 *Classification trees*

Para Hastie et al. (2009), James *et al.* (2013) e Raschka (2017), em uma árvore de classificação (*classification tree*), a predição da resposta para cada nova observação está relacionada à classe mais comum de observações da base de treinamento na região à qual ela pertence. No que tange às interpretações dos resultados, estes consistem não apenas na predição da classe correspondente a uma região de nó terminal em particular, mas também nas proporções das classes em que as observações de treinamento se enquadram.

Métodos baseados em árvore de decisão simples geram um conjunto de condições fáceis de implementar e interpretar (Kuhn & Johnson, 2013). Contudo, os algoritmos de *Machine Learning* resultantes de árvores de decisão são considerados instáveis, de modo que pequenas alterações na base de treinamento podem acarretar mudanças estruturais na árvore ou nas suas regras, o que, consequentemente, pode alterar a interpretação do modelo ajustado.

Nesse contexto, foram desenvolvidos diversos métodos a fim de melhorar o desempenho preditivo em árvores de decisão simples. Cada um desses métodos envolve a produção de múltiplas árvores, que são então combinadas para produzir uma única previsão, mais precisa, mais acurada (James *et al.*, 2013; Raschka, 2017). Dessa maneira, ante ao exposto, será discutido a seguir que a combinação de um grande número de árvores pode, muitas vezes, resultar em melhorias drásticas na acurácia da previsão, às custas de alguma perda na interpretação. Os principais métodos abordados na literatura são *bagging*, *random forests* e *boosting* (Hastie et al, 2009; James *et al.*, 2013; Kuhn & Johnson, 2013; Lantz, 2013; Izbicki & Santos, 2018).

3.7.2 *Bagging*

O *bootstrap* é o método mais abordado em situações nas quais é difícil, ou mesmo impossível, calcular diretamente o desvio padrão de uma quantidade de interesse. Em *machine learning*, o *bootstrap* pode ser usado em um contexto diferente, a fim de melhorar a performance dos métodos, representando uma técnica de reamostragem, como também pode ser adotado em conjunto com outros algoritmos que resultam em modelos instáveis, como, por exemplo, em árvores de decisão. Com o objetivo de obter uma melhor precisão na performance preditiva, o referido procedimento é conhecido como agregação *bootstrap* ou *bagging* (Hastie et al., 2009; Kuhn & Johnson, 2013).

Hastie et al. (2009) discutem que as árvores de decisão sofrem, costumeiramente, de alta variância. Por exemplo, se dividir aleatoriamente os dados da base de treinamento em duas partes e ajustar uma árvore de decisão em ambas as metades, os resultados obtidos poderão ser bem diferentes. Por outro lado, um procedimento com baixa variância produzirá resultados semelhantes se aplicado repetidamente a conjuntos de dados distintos. Nesse caso, as árvores de decisão são algoritmos que evidenciam uma significativa melhora na sua performance preditiva quando utilizadas em conjunto com o método *bootstrap* ou *bagging*.

Suponha-se um problema de classificação, e que a árvore considere um classificador $\hat{G}_b(x)$ para a classe *K* da resposta de interesse. Nesse caso, torna-se aceitável considerar uma função vetorial-indicador subjacente $\hat{f}(x)$, com valor um único e *K-1* zeros, tal que $\hat{G}(x) = \arg max_k \hat{f}(x)$. O classificador *bagging* seleciona a classe com mais "votos" das árvores *B*, $\hat{G}_{bag}(x) = \arg max_k \hat{f}_{bag}(x)$ (Hastie et al., 2009).

Logo, a previsão final da resposta de interesse para uma observação será a classe mais votada pelas *B* árvores que foram agregadas (Hastie et al., 2009). Contudo, de acordo com Kuhn e Johnson (2013), o algoritmo *bagging* apresenta uma desvantagem no que se refere ao fato de as *B* árvores agregadas evidenciarem alta correlação devido à utilização de todas as covariadas como candidatas em todas as etapas da divisão das *B* árvores de decisão. Assim, como alternativa para reduzir a correlação supracitada, o algoritmo *random forest* pode ser aplicado ao problema.

3.7.3 *Random Forest*

A abordagem inicial consistia em construir árvores inteiras com base em subconjuntos aleatórios dos

preditores (Amit & Geman, 1997; Barandiaran, 1998). Nesse sentido, Dietterich (2000) desenvolveu a ideia de seleção de divisão aleatória, em que as árvores são construídas usando um subconjunto aleatório dos principais *k*-preditores em cada divisão na árvore. Breiman (2000), por sua vez, também tentou adicionar ruído à resposta para perturbar a estrutura da árvore. Por fim, após avaliar cuidadosamente essas generalizações para o algoritmo de *bagging* original, Breiman (2000) construiu um algoritmo unificado chamado *random forest*.

O *Random Forest*, ou florestas aleatórias, proporciona uma melhoria em relação às árvores *bagging*, onde ocorre um pequeno ajuste aleatório a fim de reduzir a correlação existente entre os preditores da base de treinamento das árvores agregadas (James *et al.*, 2013). Assim como no *bagging*, descrito anteriormente, a construção do *random forest* consiste na obtenção de *B* amostras de tamanho *n* por amostragem *bootstrap* e com reposição da base de treinamento, em que, para cada base, estima-se uma árvore de decisão.

Mas, ao construir essas árvores, a cada vez que uma divisão binária recursiva é considerada, ou seja, em cada nó da árvore, uma amostra aleatória de preditores, de tamanho *m* e sem reposição, é escolhida como candidata do conjunto completo de preditores *p*. A partir de então, realiza-se, no subgrupo de amostra, a escolha da combinação preditor-ponto de corte responsável por essa segmentação (Hastie et al., 2009). O referido processo continua até que seja alcançado algum critério de parada.

É importante destacar que, na construção de uma *random forest*, a cada divisão na árvore, o algoritmo não pode sequer considerar a maioria dos preditores disponíveis. Supondo que haja um preditor muito forte na base de dados de treinamento, junto com vários outros preditores moderadamente fortes, na coleção de árvores *bagging*, a maioria ou todas as árvores usarão esse forte preditor na divisão superior. Consequentemente, todas parecerão semelhantes entre si. Portanto, as previsões das árvores *bagging* serão altamente correlacionadas. Infelizmente, calcular a média de muitas grandezas altamente correlacionadas não leva a uma redução tão grande na variância quanto a média de muitas quantidades não correlacionadas. Em suma, isso significa que o *bagging* não levará a uma redução substancial na variação de uma única árvore sob essa configuração (James *et al.*, 2013).

A estimativa da predição da resposta de interesse em *random forest* é similar à descrita no *bagging*. Para problemas de classificação, considere $\hat{G}_{rf}(x) = \arg max_k \hat{f}_{rf}(x)$ a predição da *B*-ésima árvore *random forest*, que corresponderá a determinada classe da resposta interesse, sendo baseada no voto majoritário (Hastie et al., 2009).

3.7.4 *Boosting*

Os modelos de *boosting* foram originalmente desenvolvidos para problemas de classificação e posteriormente estendidos para problemas de regressão. Surgiram no início dos anos 1990 (Schapire, 1990; Freund, 1995; Freund & Schapire, 1999), sob a influência da teoria da aprendizagem (Valiant, 1984; Kearns & Valiant, 1994). A sua ideia principal consiste em combinar a previsão de um conjunto de classificadores fracos (cuja predição é previamente melhor do que uma classificação aleatória, pois apresenta uma taxa de erro inferior) a fim de construir um classificador superior encarregado da predição final, caracterizando o método *AdaBoost*.

Segundo Kuhn e Johnson (2013), o algoritmo *AdaBoost* mostrou-se uma poderosa ferramenta de previsão, geralmente superando qualquer modelo individual. Após a incorporação, através do método *AdaBoost*, dos conceitos de função de perda, de modelagem aditiva e de regressão logística, Friedman et al. (2000) apresentaram, nas generalizações, os resultados para problemas de classificação, como também, na sua extensão, para problemas de regressão. Evidenciaram, assim, como o método *gradient boosting* tem como principal objetivo identificar um aditivo direto do modelo que minimize exponencialmente a função de perda.

Em sua forma mais simples, o *gradient boosting* baseia-se em uma dada função de perda, por exemplo SQR, e um algoritmo fraco, por exemplo árvores de regressão, dessa forma, o *gradient boosting* procura encontrar um modelo aditivo que minimize a função de perda. Em seguida, o algoritmo é inicializado com o melhor palpite da resposta de interesse, por exemplo, a média da resposta na regressão. O gradiente, por exemplo, o residual, é calculado, e um modelo é então ajustado aos resíduos para minimizar a função de perda. Assim, o modelo atual é incluído no modelo anterior, e o procedimento continua até que um número de interações especificado seja alcançado (Kuhn & Johnson, 2013).

Tanto para os problemas de regressão como para os problemas de classificação, a árvore de decisão busca subdividir o espaço dos preditores em R_j regiões diferentes, onde $j = 1, 2, \ldots J$, e prever a resposta de interesse, γ_j, para a região diferente. Dessa maneira, a regra de predição de uma árvore de decisão pode ser apresentada da seguinte forma: $x \in R_j \Rightarrow f(x) = \gamma_j$, assim como a árvore completa pode ser representada formalmente como:

$$T(x;\theta) = \sum_{j=1}^{j} \gamma_j \, I(x \, \epsilon \, R_j) \tag{10}$$

em que $\theta = \{R_j, \gamma_j\}$. As R_j regiões serão estabelecidas a partir de minimização da função perda, $L(.)$:

$$\hat{\theta} = argmin_{\theta} \sum_{i=1}^{N} L(y_i, T(x;\theta)) \tag{11}$$

Logo, a predição da j-ésima região, dada a $\hat{R}_j$, corresponderá à resposta média, para os casos de árvores de regressão, e à classe mais comum, para os casos de árvores de classificação (Hastie et al., 2009).

3.8 Critérios para avaliar o desempenho e selecionar o modelo

Está subseção tem como objetivo apresentar os principais critérios adotados para avaliar e selecionar o modelo com a melhor performance preditiva para o problema abordado, consistente com a segunda etapa do processo de ajuste dos algoritmos de ML, o aprendizado, conforme foi exposto no início da Seção 3. Como o problema adotado neste estudo tem como variável de interesse se o discente reprovou, sim ou não, isto é, uma variável qualitativa, os critérios que estão descritos a seguir dizem respeito às medidas de desempenho para os modelos de classificação, como também o teste de hipótese de McNemar para avaliar a significância das divergências entre os métodos de ML.

3.8.1 *Confusion Matrix*

De acordo com James *et al.* (2013) e Kuhn e Johnson (2013), a matriz de confusão é uma medida de desempenho para problemas de classificação, onde é apresentada a tabulação cruzada a partir de uma tabela com quatro combinações diferentes de valores reais (observados) e previstos. A Tabela 2 a seguir consiste em uma matriz de confusão para a problemática abordada nesta pesquisa, com duas possibilidades de o evento ocorrer, ou seja, quando o discente é reprovado ou não. As células da tabela indicam o número dos Verdadeiros Positivos (VP), Falsos Positivos (FP), Verdadeiros Negativos (VN) e Falsos Negativos (FN). Fica claro que as observações que se localizarem nas classes pertencentes à diagonal principal, VP e VN, são as que foram preditas corretamente, isto é, são os estudantes que foram reprovados e aprovados, respectivamente, na disciplina de Cálculo Diferencial e Integral I (James *et al.*, 2013; Kuhn & Johnson, 2013).

Tabela 2 – Matriz de Confusão para prever o risco de reprovação no ensino superior no Brasil.

		Classe Observada	
	Reprovado	**Sim (1)**	**Não (0)**
Classe Prevista	**Sim (1)**	VP	FP
	Não (0)	FN	VN

Fonte: Adaptação a partir do Livro de Kuhn e Johnson (2013).

Entretanto, as observações que se situam fora da diagonal principal correspondem aos discentes aprovados, mas que estão classificados como reprovados (FP), e os estudantes reprovados que estão classificados como aprovados (FN), e indicam os erros de classificação (James *et al.*, 2013; Kuhn & Johnson, 2013). A partir da matriz de confusão é possível determinar os critérios que foram adotados para avaliar a melhor performance preditiva dos algoritmos de ML, a saber: *Accuracy* (acurácia), *Sensitivity* (sensibilidade) e *Specificity* (especificidade).

3.8.2 *Accuracy, Sensitivity* e *Specificity*

A *Accuracy* (Acc), ou acurácia, de acordo com Kuhn e Johnson (2013), é a relação mais simples originada da matriz de confusão, e apresenta a concordância entre as classes observadas e previstas, tendo

uma interpretação mais direta. No entanto, existem algumas desvantagens: (i) primeiro, as contagens gerais de precisão não fazem distinção entre o tipo de erro cometido, ou seja, em situações em que os custos são diferentes, a precisão pode não medir as características importantes do modelo, como, por exemplo, pode classificar um aluno como reprovado quando este é aprovado ou o inverso; (ii) em segundo lugar, é preciso considerar as frequências naturais de cada classe, pois esta métrica não as considera (James *et al.*, 2013; Kuhn & Johnson, 2013). A acurácia é dada por:

$$Acc = \frac{VP+VN}{VP+FP+FN+VN} \tag{17}$$

Quando o objetivo é determinar o erro derivado de um classificador, ou seja, conhecer a performance específica de acordo com as classes da resposta de interesse (reprovado ou não), as análises de *Sensitivity* e *Specificity*, sensibilidade e especificidade, respectivamente, podem ser adotadas. A sensibilidade do modelo é a taxa das classes positivas observadas, quando é prevista corretamente. Dito de outra maneira, é a proporção de VP na classe da resposta de interesse [Sim (1)] que foi de fato observada (Kuhn & Johnson, 2013). A sensibilidade pode ser visualizada em:

$$sensitivity = \frac{VP}{VP+FN} \tag{18}$$

Kuhn e Johnson (2013) afirmam que a sensibilidade é, por vezes, considerada a taxa de verdadeiro positivo, visto que mede a precisão na população do evento. Por outro lado, a *Specificity*, ou especificidade, é definida como a proporção de VN na classe da resposta de interesse ausente [Não (0)], que também é observado, assim como na sensibilidade (James *et al.*, 2013; Kuhn & Johnson, 2013).

$$specificity = \frac{VN}{FP+VN} \tag{19}$$

Assumindo um nível fixo de precisão para o modelo, Kuhn e Johnson (2013) destacam a existência de um *trade-off* entre a sensibilidade e a especificidade. Intuitivamente, aumentar a sensibilidade de um modelo é susceptível para uma perda de especificidade, uma vez que mais amostras estão sendo previstas como eventos. Os potenciais *trade-offs* podem ser apropriados quando existem penalidades diferentes associadas a cada tipo de erro. A curva ROC (*Receiver Operating Characteristic*) é uma técnica utilizada para avaliar esse *trade-off* e é discutida a seguir.

3.8.3 A Curva *Receiver Operating Characteristic* (ROC)

O método gráfico mais comum para combinar sensibilidade e especificidade em um único valor é a curva ROC (Prati, Batista & Monard, 2008; Kuhn & Johnson, 2013). Foi projetada como um método geral que, dado um conjunto de dados contínuos, determina um limiar efetivo tal que valores acima do limiar são indicativos de um evento específico. Ressaltando-se que um algoritmo de classificação gera uma probabilidade de classes, esta ferramenta é apropriada para avaliar a sensibilidade e a especificidade decorrentes dos possíveis pontos de corte para p_k^* (Brown & Davis, 2006; Fawcett, 2006).

Destacando-se que a sensibilidade é a taxa de precisão apenas para a população do evento, e a especificidade para os não eventos, ao alterar o limite que maximiza adequadamente o compromisso entre sensibilidade e especificidade, a curva ROC só terá o efeito de tornar as amostras mais positivas (ou negativas, conforme o caso). Na matriz de confusão, ele não pode mover amostras de ambas as células da tabela fora da diagonal. Há quase sempre uma diminuição na sensibilidade ou especificidade quando 1 é aumentado (Kuhn & Johnson, 2013).

Para James *et al.* (2013), o desempenho geral de um classificador, resumido em todos os possíveis limiares, é dado pela área sob a curva (ROC) (AUC). Uma curva ROC ideal vai abraçar o canto superior esquerdo, então quanto maior a AUC, melhor o classificador. Na tentativa de simplificar a análise da curva ROC, a AUC (*Area Under the ROC Curve*) é a derivada da curva ROC, de maneira que esta busca sintetizar a curva ROC num único valor (Kuhn & Johnson, 2013), que varia de 0,0 a 1,0, tendo como o limiar de 0,5 entre elas. Isto é, superior a esse limite, o método classifica-se em uma classe e inferior na outra classe.

Logo, quanto maior a área AUC, ou seja, mais próxima de 1, melhor a performance do modelo (Meurer & Tolles, 2017).

As curvas ROC são úteis para comparar diferentes classificadores, pois levam em conta todos os possíveis limiares. Para cada variação do limiar do classificador são alteradas as taxas de VP e FP. Logo, a cada limiar candidato, a Taxa de Verdadeiro Positivo (TVP), isto é, a sensibilidade, e a Taxa de Falso Positivo, isto é, a especificidade, são representadas uma contra a outra (Kuhn & Johnson, 2013). Essas taxas nada mais são do que as contagens reais da população em cada classe. Tendo a matriz de confusão como base, as TVP e a TFP são dadas por:

$$TVP = \frac{VP}{VP + FN} \qquad TFP = \frac{FP}{VN + FP} \tag{20}$$

3.8.4 Teste de hipótese de McNemar

A fim de avaliar a distribuição dos resultados em cada modelo e o desempenho comparativo entre eles, torna-se necessário adotar um teste de hipótese. Dessa maneira, o teste não paramétrico de McNemar[9] foi adotado neste estudo para estimar a significância estatística das diferenças entre as classificações realizadas pelos algoritmos de ML, assim como em Guyon e Elisseeff (2003), Trajman e Luiz (2008), Podsiadlo e Rybinski (2016) e Chen e Hao (2017). Considerado um teste simples e confiável por Dietterich (1998), o teste de McNemar é recomendado para os casos em que os classificadores estão sendo comparados entre si.

Após uma aplicação em um teste estatístico em genética, McNemar (1947) fundamentou a operacionalização a partir de uma tabulação ou contagem de duas variáveis categóricas, denominada de tabela de contingência. O foco está em variáveis binárias, nesse caso reprovado/aprovado, para um controle de dois casos, sendo nomeada também de tabela de contingência *2x2*. O teste de McNemar tem como objetivo verificar se as divergências entre os algoritmos coincidem, ou seja, o teste não avalia a qualidade de um modelo, mas sim se os dois modelos discordam da mesma maneira ou não. Tecnicamente, é um tipo de teste de homogeneidade para tabelas de contingência.

Os dois termos adotados no cálculo do teste de McNemar capturam os erros cometidos pelos dois modelos em conjunto, de modo específico, apenas o número de discordâncias que ficam nas células não/sim e sim/não da tabela de contingência será utilizado. Se essas células apresentarem contagens semelhantes, por exemplo, o resultado mostrará que os dois modelos comentem erros de contagem na mesma proporção, apenas em instâncias diferentes do conjunto da base de teste (Dietterich, 1998).

Com uma distribuição X^2 e 1 grau de liberdade, a hipótese nula (H_0) do teste de McNemar, neste estudo, é que os algoritmos aplicados têm precisão igual para prever a reprovação dos discentes, ou, de outra forma, os classificadores têm uma proporção semelhante de erros. Por sua vez, a hipótese alternativa (H_1) é avaliar que os algoritmos têm previsões diferentes, ou, de outro modo, os classificadores têm uma proporção diferente de erros no conjunto de testes (Dietterich, 1998; Traiman & Luiz, 2008; Chen & Hao, 2017).

4. Resultados

Com o objetivo de prever o risco de reprovação dos discentes matriculados na disciplina de Cálculo Diferencial e Integral I na UFPB, nos anos de 2010 e 2016, utilizando os modelos de ML, torna-se necessário o cumprimento de três etapas essenciais: comparação, seleção e avaliação dos modelos. Dessa forma, os resultados desta pesquisa se encontram estruturados em duas subseções que objetivam identificar, a partir de diferentes abordagens, aquela que resulta em uma melhor performance preditiva.

4.1 Comparação e Seleção do Modelo

Esta subseção aborda algumas etapas fundamentais do processo de ajuste dos modelos de previsão: a comparação e a seleção dos algoritmos de classificação adotados neste estudo. O desenvolvimento do aprendizado de um modelo de previsão é avaliado a partir de um conjunto de base de teste, em que este não

[9] Para mais informações vê McNemar (1947) e Dietterich (1998).

foi utilizado no processo de ajuste dos métodos, isto é, não foi utilizado como base de treinamento. Dessa maneira, o ajustamento será feito usando uma parte dos dados, no conjunto de treinamento, para, em seguida, examinar quão bem sua previsão no restante dos dados, no conjunto de teste.

Para James *et al.* (2013), esta estrutura resultará em uma taxa de erro mais realista, de modo que o foco é o desempenho do modelo e o quanto ele é capaz de prever situações futuras dos discentes que ainda são desconhecidas. Uma vez que a base de dados (6.342 observações) se refere ao período de 2010 a 2016, optou-se por aleatorizar os dois subconjuntos no padrão 70:30, segundo Raschka (2017). Assim, 70% da base total compõe o conjunto de treinamento (4.439), e os 30% restantes formam o conjunto de teste (1.903).

Com o objetivo de adotar o conjunto de variáveis, descritas na Seção 2, que irá fornecer a melhor análise preditiva do problema em questão, o critério adotado no processo de seleção se baseou na análise de vários modelos *logit* para fins de previsões. Considerado um modelo clássico entre os algoritmos de ML, foram estimadas nove regressões logísticas com as mais diversas combinações entre os conjuntos de características. Por sua vez, a Tabela 3 apresenta todas as combinações entre as covariadas na etapa de aprendizado na base de treinamento, como também os resultados dos indicadores de qualidade (*AUC ROC*, *Accuracy*, *Sensitivity* e *Specificity*) quando os modelos já ajustados são aplicados às novas observações da base de teste na etapa de predição.

Tabela 3 – Critérios de Seleção das Variáveis - Modelo *Logit*

Dimensão	Variáveis	Modelos								
		1	2	3	4	5	6	7	8	9
Discente	Nota Vestibular	x	x	x	x	x	x	x	x	x
	Nota Vest. Mat.	x	x	x	x	x	x	x	x	x
	Casado	x	x	x	x	x	x	x	x	x
	Migrante	x	x	x	x	x	x	x	x	x
	Raça	x	x	x	x	x	x	x	x	x
	Sexo	x	x	x	x	x	x	x	x	x
	Idade Ing.	x	x	x	x	x	x	x	x	x
	Cotista	x	x	x	x	x	x	x	x	x
	Período Ing.	x	x	x	x	x	x	x	x	x
	Forma de Ing.	x	x	x	x	x	x	x	x	x
Docente	Tempo de Grad.		x	x	x	x	x	x	x	x
	Doutorado		x	x	x	x	x	x	x	x
	Public. no Ano		x	x	x	x	x	x	x	x
	Estrangeiro		x	x	x	x	x	x	x	x
	Dedic. exclusiva		x	x	x	x	x	x	x	x
	Sexo		x	x	x	x	x	x	x	x
Curso	Local do Campus			x	x	x	x	x	x	
	EF do Curso				x		x	x	x	
	EF do Centro					x	x	x		
Turma	Turno							x	x	x
	Carga Horária							x	x	x
	Média N. Vest.							x	x	x
	Média N. V. Mat.							x	x	x
	Taxa de Cotista							x	x	x
	N	1903	1903	1903	1903	1903	1903	1903	1903	1903
Critérios	AUC ROC (%)	73,03	73,17	73,17	73,52	73,68	73,52	73,56	73,56	73,16
	Accuracy (%)	66,89	67,52	67,52	67,41	67,47	67,42	67,42	67,42	67,41
	Sensitivity (%)	64,17	65,24	65,24	65,13	65,67	65,51	65,24	65,24	65,13
	Specificity (%)	69,53	69,74	69,74	69,63	69,22	69,63	69,53	69,53	69,63

Fonte: Elaboração própria a partir dos microdados do STI/UFPB e Plataforma *Lattes* do CNPq

Ante aos resultados dos critérios de desempenho expostos na Tabela 3, os modelos 7 e 8 apresentaram os conjuntos de variáveis com as melhores estimativas de previsões em uma nova base de dados: *AUC ROC* (73,56%), *Accuracy* (67,42%) e *Sensitivity* (65,24%). Ambos os modelos tiveram os mesmos resultados, muito embora as *dummies* que mensuram o *EF do Centro* variem entre eles. Sendo assim, optou-se por selecionar a base de dados que contém todas as variáveis que compõem as quatro dimensões: discente,

docente, curso e turma. Embora, no modelo de regressão logística, a covariada do *EF do Centro* não tenha impacto na análise de previsão de reprovação do discente, uma vez que não houve diferença nos resultados na sua presença ou ausência, ela pode ter influência nas estimações dos demais algoritmos de ML.

A seguir, após a determinação do conjunto de variáveis que compõem a presente análise, é necessário comparar os algoritmos e selecionar aquele com a melhor previsão relacionada à etapa de predição do processo dos modelos de ML. A Tabela 4 sumariza os resultados dos indicadores de desempenho, descritos na Subseção 3.8, adotados como critérios para selecionar o algoritmo com a melhor análise preditiva: *Accuracy*, *Sensitivity*, *Specificity* e *AUC ROC*. Por se tratar de um problema de classificação, foram estimados onze diferentes métodos e comparadas as suas respectivas performances na base de teste na etapa de predição.

Tabela 4 – Estimações dos algoritmos de *Machine Learning* para prever o risco de reprovação dos discentes matriculados na disciplina de Cálculo Diferencial e Integral I na UFPB, nos anos de 2010 e 2016.

Algoritmos		**Critérios para avaliação do desempenho (%)**				
		Accuracy	***Sensitivity***	***Specificity***	***AUC ROC***	***IC 95% (AUC)***
Penalized Methods	***Ridge***[1]	67,47	65,67	69,22	73,56	(71,34 – 75,79)
	Lasso[2]	67,20	64,81	69,53	73,02	(70,79 – 75,26)
	Elastic Net[3]	66,78	64,49	69,01	73,11	(70,88 – 75,35)
Regressão Logística		67,41	65,24	69,53	75,56	(71,34 – 75,78)
K-Nearest Neighbors (KNN)[4]		65,00	53,73	75,95	52,57	(62,01 – 66,23)
Naive Bayes Classifier		65,21	62,79	67,56	65,17	(63,04 – 67,32)
Support Vector Machines (SVM)		66,99	72,38	61,76	67,07	(64,98 – 69,17)
Decision Tree Based Methods	***C. trees***	65,73	71,10	60,51	65,81	(63,70 – 67,93)
	Bagging	68,52	62,52	70,46	74,85	(72,68 – 77,03)
	Random Forest	70,20	66,52	73,78	77,04	(74,96 – 79,13)
	Boosting	63,00	40,81	84,55	73,20	(70,97 – 75,42)

Fonte: Elaboração própria a partir dos microdados do STI/UFPB e Plataforma *Lattes* do CNPq.
Nota: [1]λ_{Ridge} = 0, 01 e MSE=0,2095; [2] λ_{Lasso} = 0, 01 e MSE=0,2113; [3]λ_{EN} = 0, 01, $\alpha = 0{,}5$ e MSE=0,2099; [4]K = 300 (Parâmetros *tunning*).

Ao analisar a Tabela 4, pode-se observar que grande parte dos algoritmos teve performances similares, com exceção dos modelos *Boosting, KNN, Naïve Bayes Classifier e Classification trees*, os quais apresentaram as menores *Accuracy* e *AUC ROC*. Como para Prati, Batista e Monard (2008) e Kuhn e Johnson (2013), a curva *ROC* é o método mais comum para combinar a *Sensitivity* e *Specificity*, o desempenho que mensura a área sob a curva *ROC*, a *AUC ROC* e a *Accuracy* foram os dois critérios, em conjunto, determinantes para selecionar os modelos com as melhores precisões de previsão: os métodos baseados em árvores de decisão: *Randon Forest e Bagging*, o *Penalized Methods Rigde*, Regressão Logística, *Penalized Methods LASSO*, *Penalized Methods Elastic Net* e o SVM.

Ao se deparar com situações como essa, Kuhn e Johnson (2013) sugerem que seja feita inicialmente a comparação dos modelos baseados em termos de performance (como na Tabela 4), ponderando alguns benefícios principais: a interpretabilidade do algoritmo, a complexidade computacional e a facilidade de implementação. Em um exemplo de escolha de modelo final, os autores supracitados destacam que os pesquisadores devem avaliar primeiro os modelos mais flexíveis e menos interpretáveis como o SVM e os *Decision tree based methodos*; em seguida, devem investigar os métodos mais simples, como o *Ridge*, o LASSO e o *Elastic Net*, isto é, os métodos de penalização. Caso ambos os modelos sejam equivalentes em termos de performance, o pesquisador deve optar pelo algoritmo mais simples que se assemelhe aos modelos mais complexos.

Logo, tendo como base as orientações de Kuhn e Johnson (2013), os modelos que deveriam ser selecionados como modelos finais neste estudo seriam: *Penalized Methods* Ridge, Regressão Logística, *Penalized Methods LASSO* e *Penalized Methods Elastic Net*. Hastie et al. (2009) indicam que os métodos lineares com penalidades são mais adequados no processo de predição, pois, de acordo com James *et al.* (2013), os algoritmos pertencentes a este grupo buscam penalizar os coeficientes estimados com o objetivo de limitar a variância ao custo de um aumento insignificante no viés. Portanto, os modelos finais selecionados, nas etapas de aprendizado (base de treinamento) e predição (base de teste), para prever o risco

de reprovação dos alunos que se matricularão na disciplina de Cálculo Diferencial e Integral I na UFPB nos próximos semestres são: *Ridge*, Regressão Logística, *LASSO* e *Elastic Net*.

Uma vez que as medidas de desempenho (*Accuracy*, *Sensitivity*, *Specificity* e *AUC ROC*) dos modelos finais selecionados, expostas na Tabela 4, são muito próximas, torna-se necessário verificar se estes são ou não estatisticamente semelhantes. Dessa maneira, para avaliar a significância estatística das diferenças entre os algoritmos de ML, foram executados os testes de McNemar de maneira individual, apresentados na Tabela 5. Os resultados dos testes de simetria X^2 e os respectivos *p-valores* estão resumidos na Tabela 5 para cada um dos modelos de ML adotados. Ao nível de significância de 5%, há evidências para não rejeitar a hipótese nula de que as distribuições das previsões sejam iguais de acordo com o *p-valor* para os seguintes algoritmos: Ridge, *LASSO*, *Elastic Net*, Regressão Logística, *Naïve Bayes Classifier*, *Bagging e Randon Forest*. No que se refere aos demais modelos, os testes evidenciam tendência para rejeitar a hipótese nula.

Tabela 5 – Testes de McNemar para os algoritmos de *Machine Learning* - Teste Individual

Algoritmos		**Teste McNemar**	
		Teste Chi-squared	*p-valor*
Penalized Methods	***Ridge***	0,9305	0,3347
	Lasso	1,9631	0,1612
	Elastic Net	1,7231	0,1893
Regressão Logística		1,5500	0,2131
K-Nearest Neighbors (KNN)		60,662	0,0000
Naive Bayes Classifier		1,8505	0,1757
Support Vector Machines (SVM)		18,919	0,0000
Decision Tree Based Methods	***C. trees***	18,222	0,0000
	Bagging	1,3088	0,2526
	Random Forest	2,1491	0,1427
	Boosting	232,99	0,0000

Fonte: Elaboração própria a partir dos microdados do STI/UFPB e Plataforma *Lattes* do CNPq.

Assim, sobre os modelos finais selecionados para prever o risco de reprovação citados anteriormente (*Ridge*, Regressão Logística, *LASSO* e *Elastic Net*), pode-se inferir, levando-se em consideração os testes de McNemar, que não há diferenças significativas nos resultados das predições de ambos os modelos. Por sua vez, para comprovar estatisticamente que os modelos finais selecionados são de fato os melhores modelos de classificação para o problema em estudo, utiliza-se novamente o teste, mas dessa vez pareando as distribuições par em par. Buscando este fim, na Tabela A.1 do apêndice, estão expostos os testes para comparação de sensibilidade (linha superior da célula) e especificidade (linha inferior da célula) para cada dois modelos de classificação, como também os respectivos *p-valores*, entre colchetes.

Pode-se observar, no se refere aos modelos finais, os quais apresentam desempenhos de predições semelhantes, que os resultados registram que não há superioridade nas medidas de sensibilidade e especificidade entre eles, isto é, não se rejeita a hipótese nula de que os algoritmos têm precisão igual para prever a reprovação dos discentes ou que os classificadores têm uma proporção semelhante de erros. Conforme pode ser observado nos pareamentos em: *Ridge* x *LASSO*; *Ridge* x *Elastic Net*; *Ridge* x Regressão Logística; *Ridge* x *Bagging*; *Ridge* x *Randon Forest*; Regressão Logística x *LASSO*; Regressão Logística x *Elastic Net*; Regressão Logística x *Bagging*; Regressão Logística x *Randon Forest*, e outros.

O contrário pode ser visto nos resultados dos testes dos modelos, por exemplo, de Regressão Logística x *Boosting*, *Ridge* x *C. trees*, *Ridge* x CVM e *Elastic Net* x KNN, os quais rejeitam a hipótese nula e indicam diferenças estatísticas entre os desempenhos. De acordo com McNemar (1947), a justificativa para o uso das medidas de sensibilidade e especificidade se dá justamente pela definição do teste, o qual busca identificar as divergências (erros) entre os algoritmos e a sua relação com os erros derivados de um classificador que são conhecidos a partir das análises de sensibilidade e especificidade.

Por fim, no que se refere à etapa de avaliação (Tabela 4), o critério da *AUC ROC*, que foi utilizada para a otimização e a seleção dos algoritmos durante o aprendizado, evidenciou desempenhos superiores a 73%, tendo a Regressão Logística o maior desempenho entre eles (75,56%). No que se refere à *Accuracy*, os quatro algoritmos previram corretamente que, em torno de 67% dos discentes na UFPB, entre os anos 2010 e 2016 (base de teste), estão em situação de reprovado e aprovado. No tocante à interpretação das

demais medidas expostas na Tabela 4, pode-se afirmar que, com uma Taxa de Falso Positivo (TFP) (1 - *Specificity*) em torno de 31% [1 - 69% (*Specificity*)] para ambos os métodos, é possível prever que 65% (*Sensitivity* ou a Taxa de Verdadeiro Positivo) dos discentes que se matricularam na disciplina de Cálculo Diferencial e Integral I na UFPB irão reprovar nos anos de 2010 e 2016, ou seja, dos 1.903 estudantes matriculados na base de tese, 1.237 foram corretamente preditos.

5 Conclusões

A reprovação dos discentes é um sério problema enfrentado pelos gestores das instituições de ensino superior, principalmente nas disciplinas que são bases e integrantes de cursos nas mais diversas áreas de ensino, como é o caso da disciplina de Cálculo Diferencial e Integral I na UFPB. Esta compõe a grade curricular de vinte e um cursos de graduação da referida universidade e varia desde centros que não exigem um conhecimento de matemática tão aprofundado, como o Centro de Ciências Sociais Aplicadas (CCSA), no curso de Ciências Atuariais por exemplo, até os centros que demandam uma base mais consistente, como o Centro de Tecnologia (TI), o Centro de Informática (CI) e outros já discutidos ao longo deste artigo.

A identificação precoce dos alunos com perfil de reprovação pode permitir que os professores, coordenadores e administradores educacionais planejem ações específicas a fim de evitar futuras reprovações em determinadas disciplinas-chave de cada curso, promovendo até um efeito em cadeia a longo prazo, como a redução dos índices de evasão e, consequentemente, a ampliação da taxa de diplomação. Tais fatos podem reduzir os custos da universidade na formação dos discentes, visto que o insucesso em uma disciplina estratégica na grade curricular (que é pré-requisito para várias outras) impacta o tempo de conclusão do curso e, portanto, aumenta o custo de oportunidade dos estudantes e, assim, estimula a evasão.

Nessa temática, esta pesquisa se propôs a aplicar algoritmos de *ML* como instrumentos para identificar o risco de reprovação dos estudantes que demandam cursar a disciplina de Cálculo Diferencial e Integral I na UFPB no período entre 2010 e 2016, com base em preditores em nível dos discentes, docentes, turma e curso, que podem contribuir na performance dos modelos.

As performances obtidas pelos 11 métodos de ML, desde os mais tradicionais (os métodos de penalização e a regressão logística) aos mais específicos (SVM e Métodos baseados em árvores) foram semelhantes. Contudo, apenas quatro deles apresentaram, em conjunto, os melhores desempenhos de previsão: *Penalized Methods* Ridge, Regressão Logística, *Penalized Methods LASSO* e *Penalized Methods Elastic*. Comparados através do teste de McNemar, não há diferenças estatisticamente significantes nas previsões entre eles, ou seja, os desempenhos entre os modelos de classificação selecionados são semelhantes. Desse modo, dos 1.903 indivíduos que compõem o conjunto de base de teste, a frequência dos alunos com *status* (reprovados e aprovados) previstos corretamente medida pela *Accuracy* foi de 67%, em ambos os modelos. Por sua vez, 65% dos discentes, referentes a 1.237 alunos, foram previstos corretamente na base de teste, medida pela *Sensitivity*, nos anos de 2010 e 2016, na disciplina de Cálculo Diferencial e Integral I na UFPB.

Por fim, a partir dos resultados encontrados no desenvolvimento da presente pesquisa, que não estão isentos de limitações, acredita-se que este é um instrumento viável para fornecer um maior suporte às ações dos gestores educacionais que visem à redução dos índices de reprovação em qualquer disciplina dos cursos de graduação de todas as instituições de ensino superior. Dessa maneira, pode-se auxiliar professores e coordenadores não só a estabelecer um debate inicial, mas também para identificar as potenciais reprovações, com o objetivo de acompanhar os alunos de maneira mais direta, buscando meios de subsidiá-los nos rendimentos acadêmicos enquanto cursam a disciplina. Consequentemente, poder-se-ia reduzir a retenção, a evasão e os custos, por um lado, e aumentar a taxa de diplomação, o estoque de mão de obra qualificada e a produtividade do ensino superior público no Brasil.

Referências

Amit, Y., & Geman, D. (1997). Shape quantization and recognition with randomized trees. *Neural computation*, *9*(7), 1545-1588.

Athey, S. (2018). The impact of machine learning on economics. In *The economics of artificial intelligence: An agenda* (pp. 507-547). University of Chicago Press.

Bajari, P., Nekipelov, D., Ryan, S. P., & Yang, M. (2015a). Machine learning methods for demand estimation. *American Economic Review*, *105*(5), 481-85.
Bajari, P., Nekipelov, D., Ryan, S. P., & Yang, M. (2015b). *Demand estimation with machine learning and model combination* (No. w20955). National Bureau of Economic Research.
Barandiaran, I. (1998). The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, *20*(8), 832-844.
Benedetti, J. K. (1977). On the nonparametric estimation of regression functions. *Journal of the Royal Statistical Society: Series B (Methodological)*, *39*(2), 248-253.
Björkegren, D., & Grissen, D. (2018). Behavior revealed in mobile phone usage predicts loan repayment. *SSRN 2611775*.
Breiman, L. (2000). Randomizing outputs to increase prediction accuracy. *Machine Learning*, *40*(3), 229-242.
Brown, C. D., & Davis, H. T. (2006). Receiver operating characteristics curves and related decision measures: A tutorial. *Chemometrics and Intelligent Laboratory Systems*, *80*(1), 24-38.
Brueckner, J. (1999). Fiscal decentralization in LDCs: the effects of local corruption and tax evasion. *Department of Economics, University of Illinois at Urbana-Champaign*.
Cavalcanti, T., Guimaraes, J., & Sampaio, B. (2010). Barriers to skill acquisition in Brazil: Public and private school students performance in a public university entrance exam. *The Quarterly Review of Economics and Finance*, *50*(4), 395-407.
Chen, Y., & Hao, Y. (2017). A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction. Expert Systems with Applications, 80, 340-355.
Cortes, C., & Vapnik, V. (1995). Support vector machine. *Machine learning*, *20*(3), 273-297.
Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. Neural computation, 10(7), 1895-1923.
Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, *40*(2), 139-157.
Diogo, M. F., dos Santos Raymund, L., Wilhelm, F. A., de Andrade, S. P. C., Lorenzo, F. M., Rost, F. T., & Bardagi, M. P. (2016). Percepções de coordenadores de curso superior sobre evasão, reprovações e estratégias preventivas. *Avaliação: Revista da Avaliação da Educação Superior*, *21*(1), 125-151.
Drucker, H., Burges, C. J., Kaufman, L., Smola, A. J., & Vapnik, V. (1997). Support vector regression machines. In *Advances in neural information processing systems* (pp. 155-161).
Econômico, 37(1), 5-39.
Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, *27*(8), 861-874.
Freund, Y. (1995). Boosting a weak learning algorithm by majority. *Information and computation*, *121*(2), 256-285.
Freund, Y., & Schapire, R. E. (1999). Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, *29*(1-2), 79-103.
Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, *28*(2), 337-407.
Goldstein, B. A., Navar, A. M., & Carter, R. E. (2017). Moving beyond regression techniques in cardiovascular risk prediction: applying machine learning to address analytic challenges. *European heart journal*, *38*(23), 1805-1814.
Gomes-Neto, J. B., & Hanushek, E. A. (1994). Causes and consequences of grade repetition: Evidence from Brazil. *Economic Development and Cultural Change*, *43*(1), 117-148.
Guimarães, J., & Sampaio, B. (2007). The influence of family background and individual characteristics on entrance tests scores of Brazilian university students. *Anais do XXXV Encontro Nacional de Economia-ANPEC-Associação Nacional dos Centros de Pós-Graduação em Economia, Recife*.
Gujarati, D. N., & Porter, D. C. (2011). *Econometria Básica-5*. Amgh Editora.
Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of machine learning research, 3(Mar), 1157-1182.
Han, J., Kamber, M. & Mining, D. (2001). Data mining: Concepts and techniques. Morgan Kaufmann, v. 340, p. 94104–3205.

Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.
Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Series in Statistics. [S.l.]: New York: springer.
Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (2018). Sinopse Estatística da Educação Superior 2017. Brasília-DF: Inep. Disponível em: <http://portal.inep.gov.br/educacao-superior>. Acesso em: 30.07.2019.
Izbicki, R., & dos Santos, T. M. (2018). Machine Learning sob a ótica estatística.
James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112, pp. 3-7). New York: springer.
Júnior, F. T., Faria, V. B., & de Lima, M. A. (2012). Indicadores de fluxo escolar e políticas educacionais: avaliação das últimas décadas. *Estudos em Avaliação Educacional*, *23*(52), 48-67.
Kearns, M., & Valiant, L. (1994). Cryptographic limitations on learning Boolean formulae and finite automata. *Journal of the ACM (JACM)*, *41*(1), 67-95.
Kleinberg, J., Mullainathan, S., & Raghavan, M. (2016). Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*.
Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling* (Vol. 26). New York: Springer.
Lantz, B. (2013). *Machine learning with R*. Packt Publishing Ltd.
Leon, F. L. L. D., & Menezes-Filho, N. A. (2002). Reprovação, avanço e evasão escolar no Brasil. Instituto de Pesquisa Econômica Aplicada (Ipea).
McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika, 12(2), 153-157.
Meurer, W. J., & Tolles, J. (2017). Logistic regression diagnostics: understanding how well a model predicts outcomes. *Jama*, *317*(10), 1068-1069.
Podsiadlo, M., & Rybinski, H. (2016). Financial time series forecasting using rough sets with time-weighted rule voting. Expert Systems with Applications, 66, 219-233.
Prati, R. C., Batista, G. E. A. P. A., & Monard, M. C. (2008). Curvas ROC para avaliação de classificadores. *Revista IEEE América Latina*, *6*(2), 215-222.
Raschka, S. (2017). *Python machine learning*. 2 ed. Birmingham: Packt Publishing Ltd.
Sampaio, B., Sampaio, Y., de Mello, E. P., & Melo, A. S. (2011). Desempenho no vestibular, background familiar e evasão: evidências da UFPE. *Economia Aplicada*, *15*(2), 287-309.
Santos, H. G. D. (2018). *Comparação da performance de algoritmos de machine learning para a análise preditiva em saúde pública e medicina* (Doctoral dissertation, Universidade de São Paulo).
Schapire, R. E. (1990). The strength of weak learnability. *Machine learning*, *5*(2), 197-227.
Shirasu, M. R., & de Albuquerque, R. (2015). Determinantes da evasão e repetência escolar no ensino médio do Ceará. *Revista Econômica do Nordeste*, *46*(4), 117-136.
Smola, A. J. et al. (1996). *Regression estimation with support vector learning machines* (Doctoral dissertation, Master's thesis, Technische Universität München).
Souza, A. P. D., Ponczek, V. P., Oliva, B. T., & Tavares, P. A. (2012). Fatores associados ao fluxo escolar no ingresso e ao longo do ensino médio no Brasil. *Pesquisa e Planejamento*
Stone, C. J. (1977). Consistent nonparametric regression. *The annals of statistics*, 595-620.
Trajman, A., & Luiz, R. R. (2008). McNemar $\chi 2$ test revisited: comparing sensitivity and specificity of diagnostic examinations. Scandinavian journal of clinical and laboratory investigation, 68(1), 77-80.
Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, *27*(11), 1134-1142.
Varian, H. R. (2014). *Big data: New tricks for econometrics. Journal of Economic Perspectives*, v. 28, n. 2, p. 3–28.
Wooldridge, J. M. (2010). *Econometric analysis of cross section and panel data*. Cambridge, Massachusetts: MIT press.