Classificação de Dados do Cadastro Ambiental Rural com uso de Algoritmos de Aprendizagem de Máquina

Fernando Elias de Melo Borges * **Danton Diego Ferreira** *
Antônio Carlos de Sousa Couto Júnior **

* *Departamento de Automática, Universidade Federal de Lavras, MG (e-mails: fborges@estudante.ufla.br; danton@ufla.br).*
** *Agência Zetta de Inovação, Universidade Federal de Lavras (e-mail: antoniocoutojr.ti@fundecc.org.br)*

Abstract: The Rural Environmental Registry (CAR) consists of a mandatory public electronic registry for all rural properties in the Brazilian territory, integrates environmental information of the properties, assists the monitoring of them, and the fight against deforestation. However, a large number of registrations are carried out erroneously generating inconsistent data, leading these to be canceled and/or to be requested to correct the registration. Carrying out these checks manually is very expensive, since a specialized workforce is required and Brazil has an immense amount of rural properties. In this context, this work aims to provide an intelligent machine learning-based system that allows to verify and classify CAR data into approved or canceled data quickly and effectively. For this purpose, three learning models were trained using real data from registers. In addition to the classification, the SMOTE tool was used to treat the imbalance between classes. Results were generated using measures of performance of classifiers and comparative studies between the methods were also performed. The results showed potential use of the method in future automated predictions, reaching performance indices above 0.90 (90%).

Resumo: O Cadastro Ambiental Rural (CAR) consiste em um registro público eletrônico obrigatório para todos os imóveis rurais do território brasileiro, integra informações ambientais das propriedades, auxilia o monitoramento das mesmas e no combate ao desmatamento. Entretanto, um grande número de cadastros é realizado de maneira errônea gerando dados inconsistentes, levando estes a serem cancelados e/ou a serem pedidas retificações para o devido preenchimento do cadastro. Realizar essas verificações de forma manual é deveras oneroso, uma vez que é requerida uma mão de obra especializada e o Brasil possui uma imensa quantidade de imóveis rurais. Neste contexto, este trabalho tem como objetivo fornecer um sistema inteligente baseado em aprendizagem de máquina que permita verificar e classificar os dados do CAR em aprovados ou cancelados de maneira rápida e eficaz. Para isto, três modelos de aprendizagem foram treinados utilizando dados reais de cadastros. Além da classificação, foi utilizada a ferramenta SMOTE para tratamento do desbalanceamento entre as classes. Foram gerados resultados utilizando medidas de desempenho de classificadores e realizados, também, estudos comparativos entre os métodos. Os resultados apresentados mostraram potencial uso do método em futuras predições automatizadas, atingindo índices de desempenho acima de 0.90 (90%).

Keywords: Rural Environmental Registry; Data Mining; Unbalanced Data; Data Classification; SMOTE

Palavras-chaves: Cadastro Ambiental Rural; Mineração de Dados; Dados Desbalanceados; Classificação de Dados; SMOTE

1. INTRODUÇÃO

Criado com o objetivo de monitorar propriedades rurais, auxiliar no combate ao desmatamento e incentivar o devido manejo sustentável das propriedades no campo, o governo brasileiro desenvolveu o Cadastro Ambiental Rural (CAR), inserido no Serviço Florestal Brasileiro (Roitman et al., 2018; Jung et al., 2017). O cadastro reúne diversas informações dos imóveis rurais no Brasil inseridas pelo cadastrante por meio do SiCAR (Sistema do Cadastro Ambiental Rural). No SiCAR, o cadastrante insere dados geográficos como a localização do terreno, área, feições do terreno (rio, vegetação existente, tipo de vegetação, por exemplo), dentre outros dados. O CAR vem auxiliando no monitoramento e também na investigação de pesquisadores sobre a influência da implementação da plataforma nas ações de desmatamento, invasões de terra, dentre outras irregularidades (L'Roe et al., 2016).

De forma a incentivar os agricultores a fazerem o cadastro e manterem o mesmo devidamente regularizado, o governo promove incentivos aos proprietários rurais que realizam o cadastro, como crédito rural facilitado com prazos e taxas melhores que o praticado no mercado, facilitação na contratação de seguro agrícola, dentre outros dispostos no Código Florestal (Brasil, 2012). Além do cadastro em si, o monitoramento geográfico das áreas ocupadas de forma a mapear as áreas degradadas e auxiliar no combate ao desmatamento se faz importante, reforçado pelos estudos realizados por dos Santos et al. (2020) e Arvor et al. (2021).

Além do monitoramento geoespacial das áreas rurais, analisar os dados do CAR de maneira a cancelar um cadastro em caso de eventual irregularidade também contribui para o monitoramento ambiental dos imóveis rurais. Entretanto, tal análise é um grande desafio, dado que o Brasil possui um grande número de imóveis rurais e analisar os cadastros de forma manual requer pessoal qualificado. Tendo em vista tal problemática, propor uma metodologia de automatizar a análise do cadastro é interessante tanto para os agricultores, que terão seus cadastros revistos de forma mais rápida, quanto para o Serviço Florestal Brasileiro.

Observando esta lacuna, este trabalho tem por objetivo propor um sistema que possa realizar a classificação dos dados do Cadastro Ambiental Rural, atribuindo a aprovação ou cancelamento dos cadastros inseridos. Para isto serão utilizadas ferramentas de aprendizagem de máquina para realizar a classificação dos dados, por meio dos algoritmos *Random Forest* (Breiman, 2001), *AdaBoost* (Hastie et al., 2009) e *Gradient Boosting* (Friedman, 2002). O uso de técnicas de *ensemble* envolvendo Árvores de Decisão, deve-se à boa capacidade de generalização destes modelos com baixo custo computacional para o treinamento. Uma vez que o conjunto de dados fornecido possui desbalanceamento, também será feito o uso de uma ferramenta de *oversampling*, por meio da geração de dados sintéticos pelo algoritmo *Synthetic Minority Over-sampling Technique* (SMOTE) (Chawla et al., 2002). A escolha do SMOTE foi motivada pela sua simplicidade de implementação e boa manutenção da distribuição original dos dados.

Técnicas de *oversampling*, como o SMOTE ou variações do mesmo, vêm sendo utilizadas de maneira a balancear o tamanho do conjunto de dados para cada classe, aumentando, assim, a recuperação desta mesma classe pelo modelo de classificação. Aplicações podem ser vistas na literatura em finanças para pontuação de microcrédito (Gicić and Subasi, 2019) e para balanceamento de dados textuais (Jonathan et al., 2020). Com relação aos algoritmos de classificação, eles possuem aplicações na literatura com bons resultados como em Uddin et al. (2020), onde os autores utilizaram o *AdaBoost* para predição de riscos de depressão em colaboradores da área de tecnologia. Outra aplicação envolvendo o *AdaBoost* é na área de segurança de redes (Shahraki et al., 2020). Aplicações envolvendo *Gradient Boosting* podem ser vistas em Sheng et al. (2018) e Dev and Eden (2019), onde o modelo foi aplicado, respectivamente, em detecção de falhas em rodovias e classificação litológica. Outro estudo que fez uso das técnicas que serão aplicadas neste trabalho pode ser observado no trabalho desenvolvido por Ge et al. (2017), no qual os autores utilizaram o SMOTE para o tratamento de desbalanceamento dos dados e o *Random Forest* como modelo de aprendizagem para predição de congelamento na superfície de pás de turbinas eólicas. Estes estudos mostram que as técnicas de aprendizagem escolhidas neste trabalho possuem boa aplicabilidade prática.

Para este trabalho serão utilizados dados reais do CAR para o desenvolvimento dos algoritmos de aprendizagem mencionados anteriormente. Os dados iniciais passarão por codificação e filtragem de variáveis. Após a geração de modelos, estes foram avaliados por meio de estudos comparativos utilizando as métricas de avaliação pertinentes aos classificadores. Além da comparação entre os classificadores, também serão realizados estudos comparativos entre os modelos treinados com e sem o uso de *oversampling*, de maneira a avaliar como o aumento da base de dados por meio da geração de dados sintéticos impactou na classificação dos cadastros.

O presente artigo segue dividido em três seções posteriores: na Seção 2 é apresentada a base de dados e os algoritmos utilizados no trabalho, bem como a sequência do desenvolvimento do mesmo; os resultados e as discussões acerca destes são descritos na Seção 3; e, por fim, as considerações finais e perspectivas de próximos passos a partir deste trabalho estão contidos na Seção 4.

2. MÉTODO PROPOSTO

2.1 Base de dados

A base de dados utilizada consiste em um compilado de variáveis existentes no CAR compostas por variáveis relacionadas ao imóvel rural, tais como: área do imóvel, número de módulos fiscais do terreno, vértices do polígono do terreno (dado como entrada em mapa desenhado na plataforma SiCAR ou por fornecimento de arquivo de georreferenciamento da propriedade); área das feições do imóvel rural, como rio, nascente, vegetações nativas (restinga, manguezal, vereda, etc.); respostas de um questionário sobre a propriedade, onde o proprietário responde perguntas sobre a regularização ambiental do imóvel rural, por meio de respostas objetivas (não, sim, não informar).

A base de dados também traz a condição do cadastro, que refere-se ao *status* do mesmo, se este encontra-se aprovado sem pendências, cancelado ou constando pendências. Como o estudo visa apenas classificar os dados em aprovados e cancelados, os cadastros pendentes não foram utilizados para análise. Portanto, como variável de saída (classe), tem-se 2 valores: 'cancelado' (0) ou 'aprovado' (1).

No total, a base de dados fornecida possui 90 atributos, sendo 89 descritores e um atributo de classe (saída) e 8012 amostras. Destas amostras, 1743 pertencentes à classe de Aprovados e 6269 pertencentes à classe de Canelados, havendo um desbalanceamento na base de dados, sendo o número de amostras da classe de Cancelados, aproximadamente, 3.6 vezes maior que o número de amostras da classe de Aprovados.

O conjunto de dados passou por uma codificação de alguns atributos textuais, como no caso do questionário, onde foi atribuído, respectivamente, -1 para 'não', 1 para 'sim' e 0 para 'não informar'. Para os cadastros que haviam

mais de um terreno inserido, logo, havendo mais de uma área no registro, foi realizada a soma destas áreas. Após a codificação, foi realizada a seleção e filtragem de variáveis. Primeiramente, foi realizada uma inspeção visual nas variáveis, verificando se algum atributo possuía somente um único valor (variável constante), excluindo-a em caso positivo. Também foi verificado que a base de dados não possui valores ausentes. Após esta inspeção, foi realizada a seleção de atributos por meio da Razão de Discriminação de Fisher (*Fisher's Discriminant Ratio* - FDR) (Duda and Hart, 2001), onde os atributos que apresentam os maiores valores de FDR são os mais relevantes para a classificação. Após a seleção das variáveis mais relevantes para a classificação, foi aplicada a normalização do tipo $z - score$ que realiza a remoção da média e a divisão pelo desvio-padrão em cada atributo da base de dados.

2.2 SMOTE

O SMOTE consiste em uma técnica de geração de dados sintéticos com base em um determinado conjunto de dados real, com a finalidade de obter um maior balanceamento de classes em problemas de classificação (Chawla et al., 2002). O algoritmo é aplicado na classe minoritária a fim de balancear o número de amostras entre as classes, de maneira a minimizar a tendência do classificador à classe majoritária.

O algoritmo possui como base o k-vizinhos mais próximos, onde ele captura os vizinhos mais próximos de cada amostra com base na distância Euclidiana entre cada amostra, ou seja, os k dados com a menor distância de uma determinada amostra. Obtidos os vizinhos mais próximos, estes são selecionados aleatoriamente de acordo com a proporção de dados a ser aumentada. Após a escolha, é calculada a diferença de cada cada atributo entre o vizinho escolhido e a amostra utilizada e gerado um valor de *gap* (valor aleatório entre 0 e 1). Os atributos dos novos dados sintéticos são obtidos pelo valor do atributo no vizinho selecionado adicionado do valor do *gap* multiplicado pela diferença calculada entre os atributos da amostra utilizada e o vizinho escolhido. Desta forma, os novos dados sintéticos são gerados.

Um diagrama em blocos do método pode ser visto na Figura 1, em que $\mathbf{X}$ é o conjunto de dados reais contendo somente a classe minoritária, N é o valor inteiro de amostras sintéticas a serem incrementadas na base de dados e k é o número de vizinhos mais próximos para busca. O procedimento apresentado pela Figura 1 é dado para uma determinada amostra i, logo, o algoritmo é executado, iterativamente, em todas as amostras do conjunto de dados reais inicialmente inserido. Assim, o algoritmo obtém novas amostras a partir do conjunto de dados real, mantendo sua distribuição e equilibrando o número de amostras por classe.

2.3 Random Forest

Floresta Aleatória (ou *Random Forest*, do inglês) é um modelo de aprendizagem por *ensemble* ou comitê. Ou seja, constituem-se por um conjunto de modelos de aprendizagem de maneira que possuam um maior poder de discriminação (Breiman, 2001). A *Random Forest* faz uso

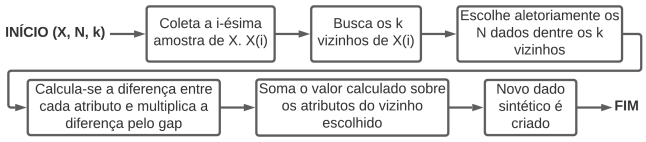

Figura 1. Diagrama em blocos do funcionamento do SMOTE.

de modelos de Árvores de Decisão, algoritmo clássico de aprendizagem de máquina, de arquitetura simples e treinamento rápido.

O modelo treina um determinado número de Árvores de Decisão de topologia especificada pelo usuário e a classificação geral do modelo é fornecida pela média das classificações de cada modelo individual. O treinamento de cada Árvore é realizado por um subconjunto do conjunto total de dados, tal subconjunto é amostrado de maneira aleatória podendo haver reposição como, por exemplo, amostragem por *bootstrapping*.

A função de margem dada pelo *Random Forest* pode ser dada por (1):

$$mg(\mathbf{X}, Y) = av_k I(h_k(X) = Y) - \max_{j \neq Y}(av_k I(h_k(\mathbf{X}) = j)), \tag{1}$$

onde h_k se refere ao k-ésimo classificador por Árvore de decisão, $\mathbf{X}$ é o conjunto de dados de entrada, Y é a saída, av_k é a média sobre o conjunto de classificadores para uma única amostra no conjunto de dados, j se refere à classe incorreta e $I(.)$ é a função indicadora, que retorna 1 se a amostra contida no vetor $\mathbf{X}$ pertence à região de classificação ou 0, caso contrário.

2.4 Algoritmos de Boosting

Baseando-se na ideia de aprendizagem por *ensemble* também presente na *Random Forest*, outros algoritmos fazem uso de múltiplos modelos de aprendizagem de menor complexidade, dentre estes, o *AdaBoost* e o *Gradient Boosting* também utilizados neste trabalho, onde todos os *ensembles* fazem uso de Árvores de decisão.

A diferença entre o algoritmo descrito na seção anterior e os dois apresentados nesta seção são o formato do comitê e a geração da classificação pelo modelo de *ensemble*. Enquanto o primeiro faz uso de amostragem e do próprio *ensemble* por meio do *bagging*, onde todos os classificadores possuem o mesmo peso na classificação final, os dois últimos fazem uso do *boosting*, onde tanto a reamostragem, quanto a classificação dos dados são feitas com pesos variáveis tanto para as amostras quanto para os modelos individuais dentro do comitê.

A regra de classificação do *ensemble* utilizando *boosting* é gerada pela média ponderada da classificação gerada pelas Árvores, onde cada uma possui um determinado peso na classificação de acordo com o erro gerado durante o treinamento, onde amostras com classificação errada têm maior peso em comparação com as amostras cujo o classificador não tem errado a predição. Em outras

palavras, o *boosting* dá maior importância aos eventos de maior dificuldade em serem preditos corretamente em detrimento da menor importância para as amostras em que o modelo não possui erros de predição.

A partir do conceito de *boosting*, métodos de treinamento de *ensembles* foram desenvolvidos, cada um com sua diferença no processamento. A principal diferença encontra-se no ajuste dos pesos para o comitê. O *AdaBoost* utiliza uma função pré-determinada para o ajuste de pesos, a depender de cada algoritmo (Hastie et al., 2009), enquanto o *Gradient Boosting* ajusta os pesos pelo método do gradiente descendente (Friedman, 2002).

O ajuste dos pesos do *ensemble* do modelo *AdaBoost* é dado pelo algoritmo SAMME.R (Hastie et al., 2009), cuja formulação é definida em (2):

$$w_i \leftarrow w_i \cdot exp\left(-\frac{K-1}{K}\mathbf{y}_i^{\intercal} log(\mathbf{p}^{(m)}(\mathbf{x}_i))\right), i = 1, ..., n, \tag{2}$$

onde w_i é o i-ésimo peso amostral para a respectiva i-ésima amostra, dentro de um conjunto de dados com n amostras, K o número de classes da base de dados, $\mathbf{y}_i$ é o valor da i-ésima saída do banco de dados de treinamento e $\mathbf{p}^{(m)}(\mathbf{x}_i)$ refere-se à probabilidade da i-ésima entrada do banco de dados $\mathbf{x}$ pertencer à determinada classe, referida à um m-ésimo classificador dentro do *ensemble*, sendo m, o índice do classificador em um comitê contendo M classificadores.

O *Gradient Boosting* ajusta seus pesos por meio de uma aproximação do gradiente descendente, tal aproximação procura minimizar uma dada função custo Ψ. Portanto, a função de ajuste dos pesos do modelo, denominado por γ_{lm} é descrita em (3):

$$\gamma_{lm} = argmin_{\gamma}\left(\sum_{\mathbf{x}_{\pi(i)} \in R_{lm}} \Psi(y_{\pi(i)}, F_{m-1}(\mathbf{x}_{\pi(i)}) + \gamma)\right), \tag{3}$$

onde $F(x)$ representa a função de decisão do modelo que realiza o mapeamento da entrada $\mathbf{x}$ com a saída y, R_{lm} é a região contendo uma subamostragem do conjunto de dados de treinamento, $\pi(i)$ é o i-ésimo valor da permutação dentro do conjunto de treinamento e os índices i e m se referem, respectivamente, ao índice das amostras da base de dados de entrada $\mathbf{x}$ e saída y e ao índice do número de classificadores contidos no *ensemble*.

2.5 Avaliação dos modelos

Após a realização do procedimento de pré-processamento, os dados foram divididos em treinamento, teste e validação. Os conjuntos de dados de treinamento e teste foram divididos em validação cruzada do tipo *k-fold* (Han et al., 2011), utilizando 10 *folds*. Já o conjunto de validação é composto por dados novos, que o modelo de aprendizagem não teve contato durante o treinamento e teste com o *k-fold*. Após o treinamento e teste com validação cruzada, foram gerados 10 classificadores, sendo o modelo que obtiver a maior AUC (área sob a curva ROC) (Han et al., 2011) escolhido para validação com os dados novos. A divisão foi feita com a proporção de 70% do número de amostras de cada classe sendo utilizadas para o treinamento e teste com validação cruzada e os 30% restantes separados para a validação na última etapa do procedimento experimental.

Como medidas numéricas de avaliação, conforme visto em Han et al. (2011), foram utilizadas, tanto para o teste em validação cruzada, quanto para a validação final, a acurácia do modelo (ACC), o *F1 - Score*, a AUC e as medidas de *precision* e *recall* para cada classe. Foram geradas as curvas ROC para cada modelo utilizando o conjunto de dados de validação e mostradas para comparativos. Os ensaios foram realizados com e sem o uso do SMOTE para fins de avaliação do impacto da ferramenta nos resultados da classificação.

3. RESULTADOS E DISCUSSÃO

Conforme apresentado na seção anterior, o conjunto inicial de dados é composto por 89 variáveis de entrada, destas, após o pré processamento, foram mantidas 49 variáveis de entrada, sendo as mais relevantes segundo o Discriminante Linear de Fisher.

Após a extração dos dados, foram realizados 2 procedimentos: o uso dos algoritmos de classificação sem o uso do SMOTE e com o uso do gerador de dados sintéticos para fins de comparação. Como parâmetros do SMOTE, a taxa de *oversampling* foi de, aproximadamente, 346% (valor para igualar o tamanho amostral entre as classes no conjunto de treinamento) e o número de vizinhos utilizado foi de $k = 17$. O valor de k foi escolhido por meio de experimentos realizados durante o projeto do método. Os tamanhos dos conjuntos de dados de treinamento e teste com validação cruzada e validação final utilizados no experimento estão contidos na Tabela 1.

Tabela 1. Conjuntos de dados de treinamento e teste com validação cruzada e validação final

Classe	Treino - Sem SMOTE	Treino - Com SMOTE	Validação
Aprovados (1)	1220	4220	523
Cancelados (0)	4220	4220	2049

Os parâmetros de cada modelo gerado para o conjunto de dados original, sem *oversampling*, encontram-se na Tabela 2, enquanto os parâmetros dos classificadores utilizando o conjunto de dados balanceado pelo SMOTE são mostrados na Tabela 3.

Tabela 2. Parâmetros dos classificadores utilizados no treinamento sem *oversampling* por SMOTE

Modelo	Parâmetro	Valor
Random Forest	Número de árvores	200
	Medida de avaliação	índice gini
	Profundidade máxima	15
AdaBoost	Número de árvores	250
	Medida de avaliação	índice gini
	Profundidade máxima	12
	Algoritmo de ajuste	SAMME.R
Gradient Boosting	Número de árvores	220
	Medida de avaliação	mse Friedman
	Profundidade máxima	5

Após os procedimentos de treinamento, teste e validação dos classificadores, os resultados numéricos e gráficos fo-

Tabela 3. Parâmetros dos classificadores utilizados no treinamento com *oversampling* por SMOTE

Modelo	Parâmetro	Valor
Random Forest	Número de árvores	200
	Medida de avaliação	entropia
	Profundidade máxima	16
AdaBoost	Número de árvores	255
	Medida de avaliação	índice gini
	Profundidade máxima	15
	Algoritmo de ajuste	SAMME.R
Gradient Boosting	Número de árvores	220
	Medida de avaliação	mse Friedman
	Profundidade máxima	7

ram gerados. Os resultados de Teste se referem aos resultados obtidos no conjunto de teste durante a validação cruzada e os resultados de validação se referem aos resultados obtidos com os dados novos, que não foram apresentados aos classificadores anteriormente. Os resultados para o conjunto de teste estão sob o formato de média ± desvio-padrão obtidos durante a validação cruzada *k-fold*. Os resultados de classificação dos dados sem o uso do SMOTE encontram-se na Tabela 4 e a respectiva curva ROC para o conjunto de validação na Figura 2. Os resultados numéricos para o conjunto de dados com *oversampling* estão contidos na Tabela 5, enquanto a sua curva ROC para o conjunto de validação está inserida na Figura 3.

Tabela 4. Resultados de desempenho para modelos treinados sem *oversampling*

Conjunto	Medida	Random Forest	AdaBoost	Gradient Boosting
Teste	ACC	0.9046 ± 0.0096	0.8954 ± 0.0114	**0.9103 ± 0.0076**
	F1 - Score	0.9380 ± 0.0063	0.9331 ± 0.0076	**0.9427 ± 0.0047**
	AUC	0.9526 ± 0.0079	0.9280 ± 0.0180	**0.9554 ± 0.0069**
	Precision - 0	**0.9452 ± 0.0116**	0.9259 ± 0.0085	0.9340 ± 0.0087
	Recall - 0	0.9313 ± 0.0128	0.9405 ± 0.0149	**0.9517 ± 0.0040**
	Precision - 1	0.7751 ± 0.0287	0.7847 ± 0.0415	**0.8211 ± 0.0132**
	Recall - 1	**0.8123 ± 0.0422**	0.7393 ± 0.0332	0.7672 ± 0.0328
Validação	ACC	0.9071	0.9016	**0.9086**
	F1 - Score	0.9408	0.9380	**0.9428**
	AUC	0.9535	0.9378	**0.9565**
	Precision - 0	**0.9557**	0.9419	0.9407
	Recall - 0	0.9263	0.9341	**0.9449**
	Precision - 1	0.7423	0.7500	**0.7802**
	Recall - 1	**0.8317**	0.7744	0.7667

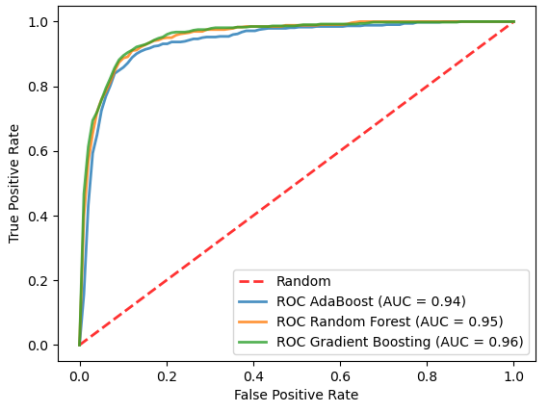

Figura 2. Curva ROC do conjunto de validação para os modelos treinados sem o uso de *oversampling* por SMOTE.

Observando as curvas ROC geradas, pode ser visto que não há melhora no desempenho global do modelo quando se usa o acréscimo de dados sintéticos no treinamento, em

Tabela 5. Resultados de desempenho para modelos treinados com *oversampling*

Conjunto	Medida	Random Forest	AdaBoost	Gradient Boosting
Teste	ACC	0.9309 +- 0.0065	0.8597 +- 0.0114	**0.9360 +- 0.0050**
	F1 - Score	0.9296 +- 0.0069	0.8586 +- 0.0112	**0.9352 +- 0.0051**
	AUC	0.9746 +- 0.0027	0.9413 +- 0.0054	**0.9809 +- 0.0037**
	Precision - 0	**0.9475 +- 0.0097**	0.8658 +- 0.0147	0.9471 +- 0.0119
	Recall - 0	0.9126 +- 0.0133	0.8517 +- 0.0141	**0.9239 +- 0.0126**
	Precision - 1	0.9158 +- 0.0113	0.8541 +- 0.0124	**0.9259 +- 0.0108**
	Recall - 1	**0.9493 +- 0.0101**	0.8678 +- 0.0166	0.9481 +- 0.0129
Validação	ACC	0.8970	0.8449	**0.9044**
	F1 - Score	0.9327	0.8969	**0.9383**
	AUC	**0.9586**	0.9235	0.9568
	Precision - 0	**0.9725**	0.9528	0.9645
	Recall - 0	0.8960	0.8472	**0.9136**
	Precision - 1	0.6886	0.5827	**0.7195**
	Recall - 1	**0.9006**	0.8356	0.8681

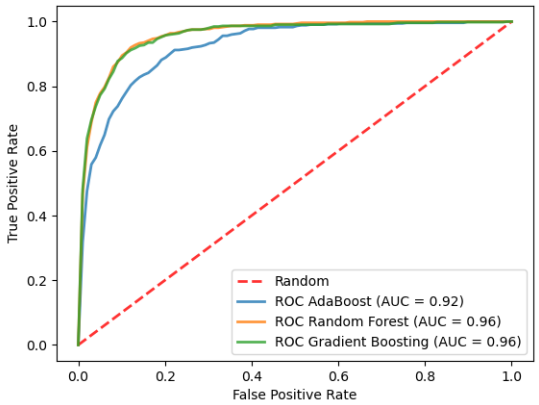

Figura 3. Curva ROC do conjunto de validação para os modelos treinados com o uso de *oversampling* por SMOTE.

específico ao *AdaBoost*, houve uma queda no desempenho. Contudo, observando os resultados das Tabelas 4 e 5 pode ser visto uma melhora significativa dos resultados na classe 1 (Aprovados). Sem o uso de *oversampling*, esta classe tem suas taxas de *recall* abaixo dos 0.85 (85%). Utilizando o SMOTE, a mesma classe tem, para o *Random Forest* e o *Gradient Boosting*, índices de *recall* acima dos 0.85, chegando a 0.90 para o primeiro algoritmo.

Cabe salientar que tal aumento do desempenho do modelo vem, conjuntamente, de um detrimento dos índices referentes à classe 0 (cancelados), visto que, para os resultados sem o uso do SMOTE, todos os valores de *recall* para a classe 0 foram acima de 0.90. Tal efeito também pode ser observado pelos índices globais de avaliação (Acurácia, AUC e *F1 - Score*), sobretudo os 2 últimos, onde o aumento do *recall* para a classe 1 não implicou em aumento das medidas globais, logo, havendo decréscimo no desempenho dos modelos para a classe 0.

A respeito do desempenho dos modelos em geral, a *Random Forest* apresentou os resultados mais consistentes e equilibrados para ambas as classes, mesmo sem *oversampling*, conseguindo recuperar a classe minoritária em valores significativamente maiores que os demais modelos. O *Gradient Boosting*, quando usado juntamente com o SMOTE apresentou melhores resultados, entretanto, sem o *oversampling* os resultados para a classe 1 foram significativamente menores. Quanto ao *AdaBoost*, os resultados atingidos foram os menores dentre os 3 modelos durante os ensaios em geral, sendo o menos recomendando para

análises posteriores, contudo, uma eventual mudança no modelo base pode ser interessante para novos testes.

Os modelos apresentados mostraram desempenhos promissores, cabendo uma nova inserção de mais dados de cadastro para melhoria do desempenho dos modelos além de eventuais ajustes nos classificadores. O SMOTE se mostrou uma técnica importante para se usar em conjunto, uma vez que, balanceando o conjunto de treinamento, os resultados se mostraram mais equilibrados entre as classes promovendo um modelo de melhor usabilidade. Tal potencial permite o avanço dos estudos no desenvolvimento de um futuro modelo de classificação dos cadastros, incrementando as análises do CAR.

4. CONCLUSÃO

Este trabalho teve como objetivo realizar uma classificação automática de dados do CAR por meio de algoritmos de aprendizagem de máquina. Devido aos dados utilizados possuírem relativo desbalanceamento entre as classes e aos resultados inicialmente gerados possuírem uma maior tendência à classe majoritária, foi incrementado o uso de um gerador de dados sintéticos (SMOTE). O uso da ferramenta de *oversampling* possibilitou uma classificação mais equilibrada entre as classes. Os resultados gerados pelos classificadores mostraram um potencial uso de algoritmos de aprendizagem de máquina para agilizar o processo de análise dos cadastros, promovendo uma análise mais rápida e assertiva. Cabe salientar, também, que um maior incremento de dados e de análises dos modelos de classificação possibilitará uma melhora nos resultados preditivos, sendo necessário para os próximos passos do trabalho.

Para trabalhos futuros, tem-se como objetivos, expandir as análises de classificação, aplicando testes estatísticos, como o teste t, avaliando detalhadamente se os valores gerados e as diferenças de classificação, como no caso do uso de *oversampling*, possuem relevância estatística. Além disto, outro objetivo é analisar as variáveis de entrada e seus impactos na classificação dos dados, para isto, utilizando métodos de aprendizagem de máquina interpretável. Permitindo, assim, uma maior visualização da tomada de decisão por parte dos analistas, promovendo uma análise rápida, eficiente e detalhada da avaliação automatizada do Cadastro Ambiental Rural.

AGRADECIMENTOS

Agradecimentos à Universidade Federal de Lavras e a Agência Zetta de inovação pelo aporte financeiro a este projeto de pesquisa.

REFERÊNCIAS

Arvor, D., Silgueiro, V., Nunes, G.M., Nabucet, J., and Dias, A.P. (2021). The 2008 map of consolidated rural areas in the brazilian legal amazon state of mato grosso: Accuracy assessment and implications for the environmental regularization of rural properties. *Land Use Policy*, 103, 105281.

Brasil (2012). Lei nº 12.651, de 25 de maio de 2012. *Diário Oficial da República Federativa do Brasil*. URL `http://www.planalto.gov.br/ccivil_03/_Ato2011-2014/2012/Lei/L12651.htm`.

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

Chawla, N.V., Bowyer, K.W., Hall, L.O., and Kegelmeyer, W.P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 321–357.

Dev, V.A. and Eden, M.R. (2019). Formation lithology classification using scalable gradient boosted decision trees. *Computers & Chemical Engineering*, 128, 392–404.

dos Santos, P.P., de Jesus Júnior, W.C., de Almeida Telles, L.A., de Souza, M.H., da Silva, S.F., dos Santos, A.R., et al. (2020). Geotechnologies applied to analysis of the rural environmental cadastre. *Land Use Policy*, 105127.

Duda, R.O. and Hart, P.E. (2001). *Pattern Classification*. John Wiely and Sons, 2 edition.

Friedman, J.H. (2002). Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4), 367–378.

Ge, Y., Yue, D., and Chen, L. (2017). Prediction of wind turbine blades icing based on mbk-smote and random forest in imbalanced data set. In *2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)*, 1–6. IEEE.

Gicić, A. and Subasi, A. (2019). Credit scoring for a microcredit data set using the synthetic minority oversampling technique and ensemble classifiers. *Expert Systems*, 36(2), e12363.

Han, J., Pei, J., and Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Hastie, T., Rosset, S., Zhu, J., and Zou, H. (2009). Multi-class adaboost. *Statistics and its Interface*, 2(3), 349–360.

Jonathan, B., Putra, P.H., and Ruldeviyani, Y. (2020). Observation imbalanced data text to predict users selling products on female daily with smote, tomek, and smote-tomek. In *2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, 81–85. IEEE.

Jung, S., Rasmussen, L.V., Watkins, C., Newton, P., and Agrawal, A. (2017). Brazil's national environmental registry of rural properties: implications for livelihoods. *Ecological Economics*, 136, 53–61.

L'Roe, J., Rausch, L., Munger, J., and Gibbs, H.K. (2016). Mapping properties to monitor forests: Landholder response to a large environmental registration program in the brazilian amazon. *Land Use Policy*, 57, 193–203.

Roitman, I., Vieira, L.C.G., Jacobson, T.K.B., da Cunha Bustamante, M.M., Marcondes, N.J.S., Cury, K., Estevam, L.S., da Costa Ribeiro, R.J., Ribeiro, V., Stabile, M.C., et al. (2018). Rural environmental registry: An innovative model for land-use and environmental policies. *Land Use Policy*, 76, 95–102.

Shahraki, A., Abbasi, M., and Haugen, Ø. (2020). Boosting algorithms for network intrusion detection: A comparative evaluation of real adaboost, gentle adaboost and modest adaboost. *Engineering Applications of Artificial Intelligence*, 94, 103770.

Sheng, P., Chen, L., and Tian, J. (2018). Learning-based road crack detection using gradient boost decision tree. In *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 1228–1232. IEEE.

Uddin, J.I., Fatema, K., and Dhar, P.K. (2020). Depression risk prediction among tech employees in bangladesh using adaboosted decision tree. In *2020 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)*, 135–138. IEEE.