Classificação e Interpretação de dados do Cadastro Ambiental Rural utilizando técnicas de Aprendizagem de Máquina

Fernando Elias de Melo Borges
Departamento de Automática
Universidade Federal de Lavras
Lavras, Minas Gerais
Email: fborges@estudante.ufla.br

Danton Diego Ferreira
Departamento de Automática
Universidade Federal de Lavras
Lavras, Minas Gerais
Email: danton@ufla.br

Antônio Carlos de Sousa Couto Júnior
Agência Zetta de Inovação
Universidade Federal de Lavras
Lavras, Minas Gerais
Email: 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. Performing automatic verification of these records is important to improve the processing of records. This paper proposes an automatic classification method to approve or cancel the CAR registers with interpretation of the classifications performed. For this, four machine learning-based classifiers were tested and the results were evaluated. The model with the best performance was used to interpret the classification using the Local Interpretable Model-agnostic Explanations (LIME) algorithm. The results showed the potential of the method in future real applications.**

***Index Terms*—Rural Environmental Registry, Data Mining, Unbalanced Data, Interpretable Machine Learning**

I. Introdução

O Cadastro Ambiental Rural (acrônimo CAR) [1] [2] é uma ferramenta do Serviço Florestal Brasileiro criada com a finalidade de realizar o monitoramento ambiental das propriedades rurais e fomentar o devido manejo sustentável da agricultura no Brasil, de maneira a prevenir e combater o desmatamento ilegal. O cadastro contém diversas informações acerca dos imóveis rurais, as quais são inseridas pelos cadastrantes por meio do sistema SiCAR. Neste sistema, são inseridos dados da propriedade, como sua área, localização, mapa do imóvel rural, dentre outros. Além disso, o CAR busca auxiliar os pesquisadores, uma vez que se trata de um registro público, a investigar a adesão ao cadastro e se este influencia nas ações de irregularidades ambientais, tais como desmatamento e invasões de terra [3].

O CAR também promove incentivos para que os proprietários dos imóveis rurais realizem o cadastro e o mantenham regularizado. Tais incentivos constam no Código Florestal [4], como exemplos destes incentivos, encontram-se facilidades no crédito rural e seguro agrícola em condições e taxas melhores que as praticadas no mercado. O CAR, além do fornecimento do cadastro e de auxiliar os produtores rurais que desejam possuir uma regularização ambiental, precisa de um monitoramento geográfico das áreas ocupadas para assegurar o devido combate ao desmatamento. Os estudos reportados em [5] e [6] mostraram a importância do monitoramento das áreas em conjunto do cadastro.

Outro ponto importante, além do monitoramento geoespacial dos imóveis rurais, são as análises do cadastro preenchido, de maneira que se possa aprovar ou cancelar os mesmos, pedindo aos cadastrantes as devidas retificações. Estas análises permitem aprovar os cadastros preenchidos corretamente e cancelar os que têm inconsistências, solicitando as devidas retificações a serem feitas no cadastro. Entretanto, analisar os cadastros manualmente é uma tarefa desafiadora, dado o grande número de imóveis rurais existentes no Brasil e da necessidade de pessoal qualificado para a realização de tal tarefa. Imerso neste desafio, o desenvolvimento de métodos para a verificação automática destes registros é de suma importância. Tal importância é aplicada tanto para o Serviço Florestal Brasileiro, que teria uma análise mais rápida e eficaz para a tomada de decisões, quanto para os agricultores que terão um retorno mais rápido da situação ambiental de suas propriedades.

Tendo em vista esta lacuna, este artigo tem como finalidade propor um sistema de verificação do Cadastro Ambiental Rural, por meio de aprendizagem de máquina, de maneira que possa aprovar ou cancelar os registros de forma automatizada. Para a realização desta tarefa, foram utilizados classificadores em conjunto com algoritmos de interpretação de modelos de aprendizagem. O procedimento envolveu as etapas de pré-processamento com o tratamento de dados desbalanceados por *oversampling* utilizando o algoritmo *Synthetic Minority Oversampling Technique* (SMOTE) [7]. Para a classificação foram implementadas Rede Neurais do tipo *Perceptron* Multicamadas (MLP, do inglês, *Multi-Layer Perceptron*) [8] e algoritmos de *ensemble*, ou máquinas de comitê, como a *Random Forest* [9], *AdaBoost* [10] e o *Gradient Boosting* [11]. Para a interpretação das decisões geradas pelo classificador, o modelo *Local Interpretable Model-agnostic Explanations* (LIME) [12] foi empregado.

O uso de *oversampling*, por meio do SMOTE ou algumas versões variadas, vêm sendo feito para balancear o tamanho do conjunto de dados para cada classe, aumentando assim a recuperação da classe minoritária, como feito no estudo reportado em [13]. Com relação aos algoritmos de classificação, estes possuem aplicações na literatura com bons resultados como em [14] e [15], que mostram estudos com o *AdaBoost* como classificador em que resultados promissores foram obtidos. Estudos envolvendo o classificador *Gradient Boosting* podem ser vistos em [16] e [17]. Numerosas aplicações de classificação de dados utilizaram Redes Neurais são encontrados na literatura, com destaque para os trabalhos de [18] e [19]. A *Random Forest* possui diversas aplicações que podem ser vistas na literatura como nos estudos realizados por [20], [21] e [22]. Exemplos na literatura envolvendo interpretação de modelos de aprendizagem de máquina por meio do LIME podem ser vistos em [23] e [24]. Os trabalhos supracitados reforçam a aplicabilidade prática dos algoritmos empregados neste trabalho.

Para este trabalho serão utilizados dados reais do CAR para o desenvolvimento dos algoritmos de aprendizagem. Para avaliar o método proposto serão utilizadas diferentes métricas de desempenho.

O presente artigo segue dividido em quatro seções. Na Seção II é apresentada uma revisão bibliográfica dos modelos utilizados neste trabalho, enquanto na Seção III é descrito o procedimento experimental realizado. Na Seção IV estão dispostos os resultados experimentais e as discussões acerca dos mesmos. As conclusões e apontamentos para os próximos passos estão inseridos na Seção V.

II. Revisão Bibliográfica

A. Synthetic Minority Over-sampling Technique - 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 [7]. O algoritmo possui como base o k-vizinhos mais próximos, onde ele captura os vizinhos mais próximos de cada amostra e, para os vizinhos escolhidos como amostra, os valores destes são adicionados de um *gap* (valor aleatório entre 0 e 1). Assim, gerando os novos dados sintéticos. Um diagrama em blocos do método pode ser visto na Figura 1. Sendo $\mathbf{X}$ um conjunto de dados reais, N o valor inteiro de aumento em dados sintéticos e k o número de vizinhos mais próximos para busca.

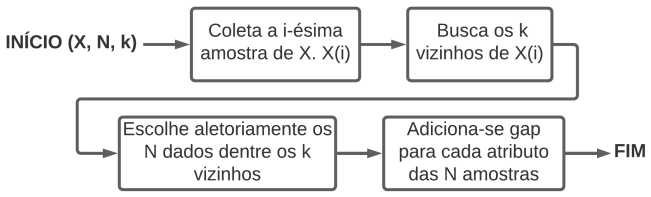

Figura 1. Diagrama em blocos do funcionamento do SMOTE.

Assim, o algoritmo obtém novas amostras a partir do conjunto de dados real, mantendo sua distribuição e equilibrando o número de amostras por classe.

B. Redes Neurais

As Redes Neurais MLP consistem em um modelo de rede neural de arquitetura *feedforward* onde os sinais de estímulo da rede são propagados somente para frente (das entradas no sentido para a saída). São modelos de boa capacidade de generalização, uma vez que fornecem um mapeamento não linear entre as entradas e saídas [8]. A função que descreve a relação entre a entrada e a saída da Rede Neural para qualquer camada k pode ser descrita na forma matricial conforme a equação (1):

$$\mathbf{u}_k = f(\mathbf{W}_k\mathbf{x} + \mathbf{b}_k) \tag{1}$$

sendo $\mathbf{u}_k$ a saída de uma camada k, $\mathbf{W}_k$ e $\mathbf{b}_k$ são, respectivamente, a matriz de pesos e o vetor de *bias* da camada k e $\mathbf{x}$ corresponde à entrada da camada k, no caso da primeira camada, $\mathbf{x}$ serão os dados de entrada da rede, nos demais casos $\mathbf{x}$ corresponde a saída da camada anterior $k-1$. A função $f(.)$ refere-se à função de ativação da rede.

O treinamento de uma Rede Neural MLP funciona a partir do algoritmo *backpropagation* [8]. Neste processo, os pesos sinápticos são ajustados no momento em que o fluxo de sinal de erro percorre contrariamente ao sinal funcional do modelo. Enquanto o sinal funcional parte dos vetores de entrada sentido à camada de saída, o sinal de erro percorre da camada de saída rumo à camada de entrada se propagando neste sentido. Desta forma, é realizado o ajuste dos pesos sinápticos da rede em conjunto com a minimização do erro do modelo de acordo com o gradiente do erro de predição.

C. 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 [9]. A *Random Forest* faz uso 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 se dá pela média das classificações de cada modelo individual. O treinamento de cada Árvore é feito 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 descrita pela equação (2):

$$mg(\mathbf{X}, Y) = av_k I(h_k(X) = Y) - \max_{j \neq Y}(avI(h_k(\mathbf{X}) = j)), \tag{2}$$

onde h_k se refere ao k-ésimo classificador por Árvore de decisão, $\mathbf{X}$ é o conjunto de dados de entrada, Y a saída e $I(.)$ a função indicadora.

D. 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 supracitado na subseção anterior e os dois apresentados nesta subseção se dá pelo 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 é feita 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 feita pelas Árvores de Decisão, onde cada uma possui um determinado peso na classificação de acordo com o erro de predição calculado 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 aos quais 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. Onde o *AdaBoost* utiliza uma função pré-determinada para o ajuste de pesos, a depender de cada algoritmo [10], enquanto o *Gradient Boosting* ajusta os pesos pelo método do gradiente descendente [11].

Um dos algoritmos de ajuste de peso do *AdaBoost* é o SAMME.R. Sua função de ajuste é descrita pela equação (3):

$$w_i \leftarrow w_i \cdot exp\left(\frac{K-1}{K}\mathbf{y}_i log(\mathbf{p}^{(m)}(\mathbf{x}_i))\right), i = 1, ..., n, \quad (3)$$

onde w_i é o i-ésimo peso amostral para a respectiva i-ésima amostra, 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 $\mathbf{p}(\mathbf{x}_i)$ refere-se à probabilidade da i-ésima entrada do banco de dados x pertencer à determinada classe.

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 pela equação (4):

$$\gamma_{lm} = argmin_\gamma \left(\sum_{\mathbf{x}_{\pi(i)} \in R_{lm}} \Psi(y_{\pi(i)}, F_{m-1}(\mathbf{x}_{\pi(i)}) + \gamma)\right), \quad (4)$$

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 e $\pi(i)$ é o i-ésimo valor da permutação dentro do conjunto de treinamento.

E. Local Interpretable Model-agnostic Explanations - LIME

O LIME [12] é um modelo de interpretação local de algoritmos de aprendizagem de máquina do tipo *Model-agnostic*, ou seja, não necessita de informações do modelo de aprendizagem desenvolvido, apenas da saída do mesmo. O LIME tem como o princípio o uso de um modelo preditivo local de baixa complexidade e boa interpretabilidade para explicar modelos de aprendizagem mais complexos.

O princípio de funcionamento do LIME consiste na obtenção da saída de uma dada entrada fornecida e, a partir deste ponto de saída, são geradas perturbações ao entorno deste ponto. A partir deste pequeno conjunto de dados composto pela entrada, a saída predita e suas respectivas perturbações, é ajustado um modelo local simplificado.

Após a obtenção do modelo local, o LIME realiza a interpretação por meio de gráfico ou tabela, indicando o intervalo de validade de cada variável para a interpretação e o valor do peso de cada variável e seus respectivos sinais. O sinal do peso indica se esta variável contribui para o aumento ou redução da saída para um problema de regressão. Em problemas de classificação, o sinal do peso indica se contribui para o aumento ou redução da probabilidade da observação pertencer à determinada classe.

Em [12] também é proposta uma forma de analisar um determinado conjunto de amostras, trata-se do algoritmo *Submodular Pick* realiza várias interpretações locais em conjunto com um algoritmo de otimização que aproxima a função que maximiza a cobertura das componentes mais importantes para a interpretação do modelo de aprendizagem. Esta função de maximização é determinada pela equação (5):

$$Pick(W, I) = \arg \max_{V, |V| \leq B} c(V, \mathbf{W}, I) \quad (5)$$

onde c é a função de cobertura das importâncias, B o número de explicações que o usuário deseja, V o conjunto de dados, $\mathbf{W}$ corresponde à matriz de explicação e I computa a importância total das variáveis.

III. Método Proposto

A. Base de dados e pré-processamento

A base de dados obtida consiste em um compilado de variáveis existentes no CAR compostas por informações relacionadas ao imóvel rural, tais como: área do imóvel; número de módulos fiscais da propriedade; vértices do polígono do terreno (dado como entrada em mapa desenhado no módulo de cadastro do CAR ou por fornecimento de arquivo de georreferenciamento da propriedade); área das feições do imóvel rural, como rio, nascente, vegetações nativas (resinga, manguezal, vereda, etc.); e respostas de um questionário sobre a propriedade, onde o proprietário responde acerca do imóvel

perguntas sobre a regularização ambiental do imóvel rural por meio de respostas objetivas (não, sim, não informar).

Além das informações supracitadas, também é fornecida a condição do cadastro, que refere-se ao *status* do mesmo, se o registro se encontra 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. Outro ponto a ser salientado, é que todos os cadastros utilizados foram coletados uma única vez, de maneira que não haja duplicatas entre os cadastros (dois cadastros iguais de versões diferentes).

No total, a base de dados fornecida possui 91 atributos, sendo 90 descritores e um atributo de classe (saída), sendo todos os atributos preenchidos e, portanto, sem valores ausentes. O conjunto de dados passou por uma codificação de alguns atributos textuais (para o caso do questionário) ou para a soma de cadastros com mais de uma área, por exemplo. A base passou por procedimentos de pré-processamento como remoção de variáveis não pertinentes, análise visual e seleção de atributos, esta última feita por meio do Discriminante Linear de Fisher [25]. Após a seleção de atributos, foi observada a matriz de correlação entre as variáveis e, para o caso de variáveis redundantes (com alta correlação entre si), foi realizada a remoção das variáveis redundantes. Após a seleção dos atributos, estes passaram por normalização do tipo *z-score* (normalização que realiza a remoção da média e divisão das amostras pelo desvio padrão de cada atributo).

B. Avaliação dos classificadores e uso do interpretador

Após a realização do pré-processamento dos dados, foram projetados os modelos de classificação dos dados do CAR. Antes do projeto dos classificadores, o SMOTE foi aplicado à classe minoritária para o balanceamento do número de amostras por classe no conjunto de treinamento. Gerando, assim, dois conjuntos de dados para treinamento dos classificadores: o conjunto de dados original e o conjunto de dados superamostrado pelo SMOTE. O conjunto de teste foi o mesmo para ambos os casos. A partir dos modelos desenvolvidos, os resultados foram obtidos e os ensaios comparativos foram realizados entre os algoritmos de aprendizagem utilizados neste trabalho.

Cada modelo foi treinado sob validação cruzada do tipo *k-fold* com 10 *folds* e o modelo a ser escolhido para teste foi o modelo de maior AUC (área sob a curva ROC) calculada, logo, o modelo pertencente ao *fold* de maior AUC. Como medidas numéricas de avaliação, foram utilizadas, tanto para treino quanto para teste, a acurácia do modelo (ACC), a AUC e as medidas de *precision* e *recall* para cada classe. Foram geradas as curvas ROC para cada modelo 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.

Finalizados os testes comparativos, o modelo de melhor desempenho geral será escolhido para ensaios no LIME. Será utilizado o *Submodular Pick* para realizar uma análise geral, do conjunto de dados de teste para analisar quais variáveis tem maior impacto em cada classe, além do intervalo no qual estas possuem este impacto na saída. Desta forma, será possível elencar quais variáveis seriam mais influentes na tomada de decisão do modelo de aprendizagem. Permitindo a observação de quais mais influenciam na aprovação ou reprovação do cadastro.

IV. Resultados e Discussão

A. Resultados dos testes comparativos entre os classificadores

Após os procedimentos de pré-processamento, o número de variáveis foi reduzido para 42 (41 entradas e 1 saída). Em seguida, foi aplicado o SMOTE para geração da superamostragem da classe minoritária. O parâmetro k do SMOTE, referente ao número de vizinhos mais próximos, foi ajustado diversas vezes para escolha do parâmetro que gerou os melhores resultados ($k = 9$). Na Tabela I estão contidos os valores amostrais com e sem o uso do SMOTE.

Tabela I
Conjuntos de dados de treinamento e teste

Classe	Treino - Sem SMOTE	Treino - Com SMOTE	Teste
Aprovados - 1	1394	5015	349
Cancelados - 0	5015	5015	1254
Total	6409	10030	1603

Realizados os procedimentos anteriormente mencionados, foi feito o projeto dos classificadores. Os hiperparâmetros de cada modelo sem e com o uso do SMOTE para superamostragem seguem, respectivamente, nas Tabelas II e III.

Tabela II
Hiperparâmetros dos modelos treinados com o conjunto de dados original

Modelo	Parâmetro	Valor
Rede MLP	Camadas ocultas	1
	Neurônios nas camadas ocultas	100
	Função de ativação das camadas ocultas	ReLu
	Neurônios na camada de saída	2
	Função de ativação da camada de saída	softmax
	Algoritmo de otimização	Quasi-Newton (2ª ordem)
	Taxa de aprendizagem	adaptativa
Random Forest	Número de árvores	500
	Medida de avaliação	índice Gini
	Profundidade máxima	12
AdaBoost	Número de árvores	200
	Medida de avaliação	Entropia
	Profundidade máxima	12
	Algoritmo de ajuste	SAMME.R
Gradiet Boosting	Número de árvores	200
	Medida de avaliação	mse_friedman
	Profundidade máxima	10

Após o treinamento e teste dos algoritmos de aprendizagem, foram gerados os resultados de classificação tanto para o *dataset* original, quanto para os dados superamostrados pelo SMOTE. Os resultados numéricos para o conjunto de dados original estão contidos na Tabela IV, enquanto os resultados para o conjunto com uso de *oversampling* seguem na Tabela V. Os resultados para o treinamento estão no formato média $\pm$ desvio padrão e os resultados de teste são os resultados do modelo escolhido durante a validação cruzada (modelo que obteve maior AUC no *fold* de validação). Além dos resultados

Tabela III
HIPERPARÂMETROS DOS MODELOS TREINADOS COM O CONJUNTO DE DADOS SUPERAMOSTRADO

Modelo	Parâmetro	Valor
Rede MLP	Camadas ocultas	1
	Neurônios nas camadas ocultas	100
	Função de ativação das camadas ocultas	ReLu
	Neurônios na camada de saída	2
	Função de ativação da camada de saída	softmax
	Algoritmo de otimização	Quasi-Newton (2ª ordem)
	Taxa de aprendizagem	adaptativa
Random Forest	Número de árvores	400
	Medida de avaliação	índice Gini
	Profundidade máxima	14
AdaBoost	Número de árvores	500
	Medida de avaliação	Entropia
	Profundidade máxima	12
	Algoritmo de ajuste	SAMME.R
Gradiet Boosting	Número de árvores	500
	Medida de avaliação	mse_friedman
	Profundidade máxima	16

numéricos, foram geradas as curvas ROC para os dados de teste. A curva ROC para o conjunto de dados original é mostrada na Figura 2, enquanto a curva ROC para o conjunto de dados superamostrado é apresentada na Figura 3.

Tabela IV
RESULTADOS DE DESEMPENHO DOS MODELOS DE APRENDIZAGEM COM TREINAMENTO REALIZADO COM OS DADOS ORIGINAIS

Conjunto	Medida	Rede MLP	*AdaBoost*	*Random Forest*	*Gradient Boosting*
Treino	ACC	0,8588 ± 0,0163	0,8969 ± 0,0118	0,8889 ± 0,0084	**0,9067 ± 0,0092**
	F1 - Score	0,9116 ± 0,0099	0,9350 ± 0,0073	0,9273 ± 0,0056	**0,9410 ± 0,0058**
	AUC	0,8759 ± 0,0155	0,9296 ± 0,0152	0,9467 ± 0,0101	**0,9497 ± 0,0065**
	Precision - 1	0,7061 ± 0,0420	0,7921 ± 0,0302	0,7104 ± 0,0194	**0,8119 ± 0,0286**
	Recall - 1	0,6005 ± 0,0544	0,7137 ± 0,0458	**0,8271 ± 0,0287**	0,7446 ± 0,0349
	Precision - 0	0,8935 ± 0,0134	0,9227 ± 0,0113	**0,9497 ± 0,0079**	0,9307 ± 0,0087
	Recall - 0	0,9306 ± 0,0101	0,9478 ± 0,0088	0,9061 ± 0,0087	**0,9517 ± 0,0089**
Teste	ACC	0,8440	0,9014	0,9008	**0,9127**
	F1 - Score	0,9024	0,9374	0,9350	**0,9447**
	AUC	0,8597	0,9375	**0,9548**	0,9515
	Precision - 1	0,6678	0,7851	0,7317	**0,8215**
	Recall - 1	0,5645	0,7536	**0,8596**	0,7650
	Precision - 0	0,8838	0,9322	**0,9589**	0,9358
	Recall - 0	0,9219	0,9426	0,9123	**0,9537**

Tabela V
RESULTADOS DE DESEMPENHO DOS MODELOS DE APRENDIZAGEM COM TREINAMENTO REALIZADO COM OS DADOS SUPERAMOSTRADOS PELO SMOTE

Conjunto	Medida	Rede MLP	*AdaBoost*	*Random Forest*	*Gradient Boosting*
Treino	ACC	0,8847 ± 0,0138	**0,9258 ± 0,0077**	0,9235 ± 0,0101	0,9222 ± 0,0070
	F1 - Score	0,8774 ± 0,0166	**0,9247 ± 0,0077**	0,9221 ± 0,0105	0,9211 ± 0,0070
	AUC	0,9350 ± 0,0091	0,9712 ± 0,0039	0,9733 ± 0,0040	**0,9749 ± 0,0053**
	Precision - 1	0,8457 ± 0,0228	**0,9135 ± 0,0091**	0,9084 ± 0,0122	0,9109 ± 0,0082
	Recall - 1	**0,9424 ± 0,0105**	0,9408 ± 0,0121	0,9422 ± 0,0105	0,9362 ± 0,0125
	Precision - 0	0,9351 ± 0,0103	0,9391 ± 0,0117	**0,9400 ± 0,0107**	0,9345 ± 0,0119
	Recall - 0	0,8271 ± 0,0308	**0,9109 ± 0,0102**	0,9049 ± 0,0134	0,9083 ± 0,0093
Teste	ACC	0,8222	0,8821	**0,9039**	0,8890
	F1 - Score	0,8786	0,9225	**0,9364**	0,9270
	AUC	0,8775	0,9336	**0,9569**	0,9461
	Precision - 1	0,5627	0,6914	**0,7241**	0,7041
	Recall - 1	0,8223	0,8281	**0,9026**	0,8453
	Precision - 0	0,9433	0,9494	**0,9709**	0,9544
	Recall - 0	0,8222	0,8971	**0,9043**	0,9011

Após a geração dos resultados de classificação, alguns pontos importantes podem ser destacados em relação ao uso do SMOTE para superamostragem: (i) o uso de *oversampling* pelo SMOTE proporcionou um equilíbrio entre as classes durante o treinamento e tal balanceamento viabilizou uma classificação menos tendenciosa à classe majoritária; (ii) os índices tornaram-se mais equilibrados, entretanto, conforme pode ser visto comparando os resultados das Tabelas IV e V, o desempenho para a classe majoritária sofreu reduções após

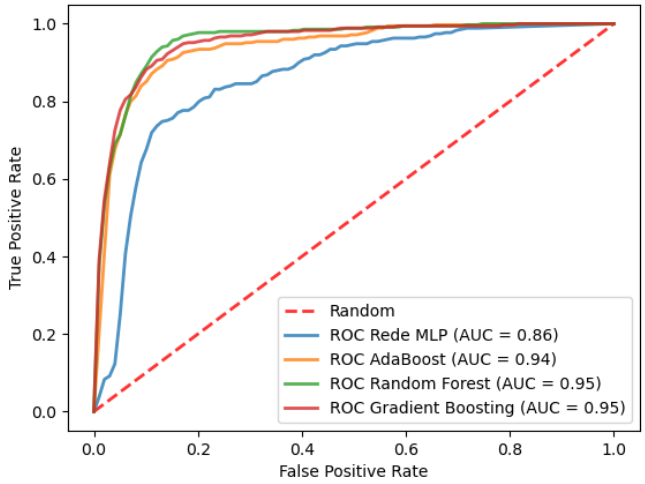

Figura 2. Cuva ROC par modelos treinados com *dataset* original.

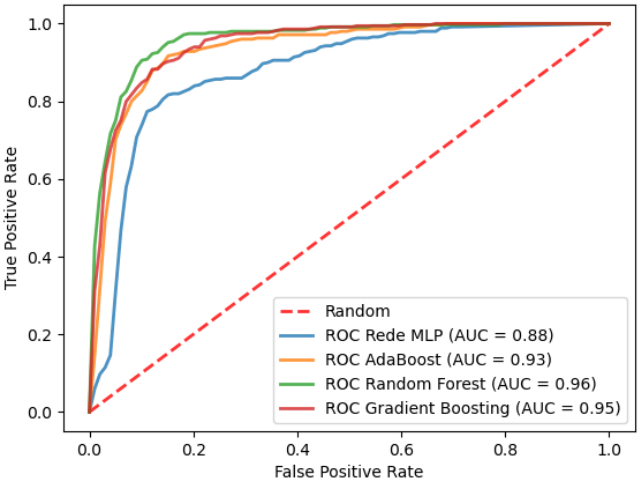

Figura 3. Cuva ROC par modelos treinados com *dataset* superamostrado.

o balanceamento entre as classes; (iii) Quanto aos índices de desempenho globais, como ACC, AUC e F1 - Score, o uso do superamostrador não gerou alterações significativas.

A respeito do desempenho geral dos classificadores, visto tanto pelas Tabelas IV e V, quanto pelas curvas ROC das Figuras 2 e 3, foi possível analisar um modelo mais viável para os testes no LIME. Dentre os quatro modelos utilizados, a Rede MLP apresentou o desempenho mais baixo no geral, em comparativo com os demais modelos, podendo ser visto pela sua curva ROC significativamente de menor área que as demais. Os classificadores *AdaBoost* e *Gradient Boosting* apresentaram índices superiores à Rede Neural, contudo, os resultados apresentados por estes dois modelos, cujo desempenho foi similar, apresentou ligeira tendência para a classe de Cancelados, além dos índices apresentarem valores iguais ou inferiores aos resultados gerados pela *Random Forest*. O classificador que obteve o melhor desempenho geral foi a *Radom Forest* que apresentou os melhores resultados gerais, dando enfoque ao teste. Os resultados em teste utilizando

a *Random Forest*, sobretudo quando treinada com o *dataset* balanceado, se mostraram superiores, em maior ou menor diferença, aos demais modelos. Portanto, este último modelo foi selecionado para as posteriores análises de interpretação pelo LIME.

B. Resultados de interpretação utilizando o LIME

Realizados os testes comparativos entre os classificadores e selecionando o modelo de melhor desempenho geral, foram realizadas as análises de interpretação por meio do LIME. Para obter uma interpretação de todo o conjunto de teste, foi utilizada a versão *Submodular Pick* do LIME, de forma a ser observada uma relação entre as variáveis de entrada com a saída de maneira global. Durante os testes, a largura do *kernel* foi ajustada de maneira que a predição local seja compatível com a predição do classificador sem sobreajuste do modelo de interpretação. O valor de *kernel* utilizado empiricamente foi de $1,5$.

Como resultados gerados, o LIME retornou 72 interpretações do classificador em relação aos dados de teste. De maneira a gerar uma visualização compactada, contendo as interpretações de maior impacto, foi gerado o gráfico de interpretação por classe dos 20 intervalos de interpretação com maior peso. O gráfico que mostra as explicações geradas pelo LIME é apresentado na Figura 4.

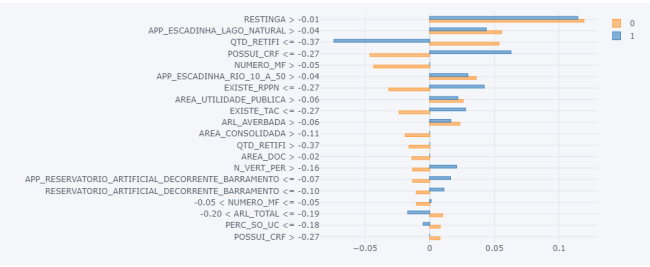

Figura 4. Gráfico de ranqueamento por classe das predições locais geradas pelo LIME. Onde '0' se refere à classe de dados Cancelados e '1' se refere à classe de Aprovados.

A explicação por classe, como mostrada na Figura 4, apresenta os pesos relativos para cada classe juntamente com seu sinal. O sinal negativo indica que o intervalo da predição descrita pelo LIME impacta negativamente na probabilidade de determinada amostra pertencer àquela classe e vice-versa. Um exemplo é a variável 'POSSUI_CRF', onde esta possui um impacto positivo para o a classe de Aprovados e negativo para a classe de Cancelados, logo, pelo intervalo da predição, se esta variável for menor ou igual à $-0,27$ (em valores normalizados pelo *z-score*), ela influenciará em redução da probabilidade da amostra ser classificada com Cancelada e terá um aumento na probabilidade da mesma amostra ser classificada com Aprovada.

Com relação às interpretações geradas, alguns pontos cabem ser destacados, como: (i) as interpretações possibilitam uma observação geral de como cada variável impacta na tomada de decisões e em favor de qual classe, contudo, análises posteriores se fazem necessárias, com a finalidade de confrontar as análises geradas pelo LIME com as análises reais por parte especialista do CAR; (ii) algumas explicações geradas podem gerar interpretações com ambiguidade, como o caso da variável 'RESTINGA', onde a interpretação gerada pelo LIME aponta que esta variável impacta positivamente para ambas as classes. Tal ambiguidade se deve ao fato de não ser possível discriminar a influência da variável em cada classe.

Outro ponto a ser salientado são os valores apresentados no intervalo, estes em valores normalizados pelo *z-score*. Logo, para uma análise mais consistente, se faz necessário a conversão do valor normalizado para o valor real da variável. A visualização pelo valor real permite observar inconsistências na interpretação, como uma área negativa, por exemplo. Um caso a ser mostrado é a variável 'AREA_DOC', onde seu intervalo convertido para o valor real seria de $-228,42$. Logo, sendo um valor de área negativa, pode gerar uma interpretação inconclusiva. Contudo, as análises geradas, se bem ajustadas e condizentes com a aplicação, permitem uma visualização ampla da tomada de decisão automática, auxiliando os analistas do CAR. Tendo em vista tal fator, novos estudos se fazem necessários de maneira a melhorar as análises de interpretação pelo LIME.

V. Conclusão

Este trabalho teve como objetivo propor um método de classificação e análise automatizada do CAR, fornecendo, além da predição do cadastro, aprovando-o ou cancelando-o, uma visualização do impacto de cada variável na tomada de decisão feita pelo classificador. Além do processamento dos dados pelos classificadores, este trabalho cobriu todo o pré-processamento, incluindo o tratamento de dados desbalanceados por meio de algoritmo superamostrador. Os resultados gerados mostraram potencial de aplicação do método, cabendo melhorias, sobretudo no método de interpretação, comparando as interpretações geradas com as análises práticas manuais e tratando eventuais explicações inconsistentes geradas pelo LIME.

Para trabalhos futuros, tem-se por objetivos o aumento do número de cadastros a serem utilizados na aplicação do método, à medida em que novos cadastros forem rotulados pela equipe especialista. Além do aumento amostral, outro objetivo é o incremento de variáveis geoespaciais para aprimoramento da classificação. Outro ponto a ser analisado posteriormente é a visualização do impacto das classificações erradas geradas pelos modelos de aprendizagem e como tratar tal situação, podendo, ou não, flexibilizar o classificador para uma determinada classe. Além da classificação, serão realizadas maiores análises na interpretação das tomadas de decisão, testes com outros algoritmos de interpretação de classificadores, fazendo um comparativo com a interpretação gerada pelo LIME, além da comparação dos resultados com a prática real antes da implementação do método no CAR. Tais melhorias tendem a aprimorar o método de forma a fornecer uma análise assertiva, rápida e eficiente, auxiliando os analistas nas tomadas de decisão e fornecendo um retorno rápido do cadastro para os agricultores brasileiros.

AGRADECIMENTOS

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

REFERÊNCIAS

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

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

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

[4] Brasil, "Lei nº 12.651, de 25 de maio de 2012," *Diário Oficial da República Federativa do Brasil*, 2012. [Online]. Available: http://www.planalto.gov.br/ccivil_03/_Ato2011-2014/2012/Lei/L12651.htm

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

[6] D. Arvor, V. Silgueiro, G. M. Nunes, J. Nabucet, and A. P. Dias, "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*, vol. 103, p. 105281, 2021.

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

[8] S. Haykin, *Redes neurais: princípios e prática*. Bookman Editora, 2007.

[9] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[10] T. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class adaboost," *Statistics and its Interface*, vol. 2, no. 3, pp. 349–360, 2009.

[11] J. H. Friedman, "Stochastic gradient boosting," *Computational statistics & data analysis*, vol. 38, no. 4, pp. 367–378, 2002.

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

[13] A. Gicić and A. Subasi, "Credit scoring for a microcredit data set using the synthetic minority oversampling technique and ensemble classifiers," *Expert Systems*, vol. 36, no. 2, p. e12363, 2019.

[14] J. I. Uddin, K. Fatema, and P. K. Dhar, "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)*. IEEE, 2020, pp. 135–138.

[15] A. Shahraki, M. Abbasi, and Ø. Haugen, "Boosting algorithms for network intrusion detection: A comparative evaluation of real adaboost, gentle adaboost and modest adaboost," *Engineering Applications of Artificial Intelligence*, vol. 94, p. 103770, 2020.

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

[17] V. A. Dev and M. R. Eden, "Formation lithology classification using scalable gradient boosted decision trees," *Computers & Chemical Engineering*, vol. 128, pp. 392–404, 2019.

[18] G. Karimi and M. Heidarian, "Facial expression recognition with polynomial legendre and partial connection mlp," *Neurocomputing*, vol. 434, pp. 33–44, 2021.

[19] J. P. L. Pizzaia, I. R. Salcides, G. M. de Almeida, R. Contarato, and R. de Almeida, "Arabica coffee samples classification using a multilayer perceptron neural network," in *2018 13th IEEE International Conference on Industry Applications (INDUSCON)*. IEEE, 2018, pp. 80–84.

[20] B. Kalaiselvi and M. Thangamani, "An efficient pearson correlation based improved random forest classification for protein structure prediction techniques," *Measurement*, vol. 162, p. 107885, 2020.

[21] P. Kumar, G. G. Nair *et al.*, "An efficient classification framework for breast cancer using hyper parameter tuned random decision forest classifier and bayesian optimization," *Biomedical Signal Processing and Control*, vol. 68, p. 102682, 2021.

[22] M. J. Sagayaraj, V. Jithesh, and D. Roshani, "Comparative study between deep learning techniques and random forest approach for hrrp based radar target classification," in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*. IEEE, 2021, pp. 385–388.

[23] B. VanBerlo, M. A. Ross, J. Rivard, and R. Booker, "Interpretable machine learning approaches to prediction of chronic homelessness," *Engineering Applications of Artificial Intelligence*, vol. 102, p. 104243, 2021.

[24] S. Sahay, N. Omare, and K. Shukla, "An approach to identify captioning keywords in an image using lime," in *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*. IEEE, 2021, pp. 648–651.

[25] R. O. Duda and P. E. Hart, *Pattern Classification*, 2nd ed. John Wiely and Sons, 2001.