ISSN 0000-0000 / e-ISSN 1981-1004

Embrapa Uva e Vinho

Boletim de Pesquisa e Desenvolvimento 36

Bento Gonçalves, RS / Maio, 2025

Modelo de aprendizado supervisionado para validações a campo do algoritmo Embrapa/MAHM

Fabio Rossi Cavalcanti

Pesquisador, Embrapa Uva e Vinho, Bento Gonçalves, RS.

Resumo – Recentes avanços tecnológicos vêm levando ao desenvolvimento e implantação de inteligência artificial, algoritmos para detecção precoce e previsão de doenças em plantas. No presente trabalho, foi desenvolvido um modelo de classificação em aprendizado de máquina (*Machine Learning*) utilizando um classificador por votação que combinou três algoritmos: *Support Vector Classifier* (SVC), Regressão Logística e *Random Forest*. O objetivo foi reproduzir os alertas de pulverização gerados pelo método "GMAHM" do algoritmo Embrapa/MAHM, para o controle do míldio da videira. Utilizou-se o *framework Scikit-Learn do Python* em conjuntos de dados com 1.200 linhas e 168 características ambientais, além de uma coluna adicional com os alertas a serem classificados. A sequência de trabalho incluiu: tratamento dos dados e definição das características e da variável alvo, normalização das características e conversão numérica das variáveis alvo, divisão dos dados em conjuntos de treino e teste, treinamento dos modelos individuais e combinação em um classificador por votação, configuração e avaliação do *pipeline* com validação cruzada, otimização de hiperparâmetros, e avaliação do modelo otimizado no conjunto de teste e predições com novas características. Embora o modelo combinado tenha atingido uma precisão de 97% no treinamento/teste, as predições com dados novos foram menores (84%), apresentando leve superestimação. Mesmo assim, o modelo se apresenta como candidato eficaz para treinamento com dados de observações a campo em trabalhos de validação, para versões mais acuradas do Embrapa/MAHM.

Termos para indexação: aprendizado de máquina, máquinas de vetores de suporte, regressão logística, floresta aleatória, ciência de dados.

Supervised machine learning model for field validations of the algorithm Embrapa/MAHM

Abstract – Technological advancements have led to the development and deployment of artificial intelligence algorithms for the early detection and prediction of plant diseases. A classification model in Machine Learning was developed using a voting classifier that combined three algorithms: Support Vector Classifier (SVC), Logistic Regression, and Random Forest. The objective was to reproduce the spray alerts generated by the 'GMAHM' method of the Embrapa/MAHM alert algorithm for controlling grapevine downy mildew. The Scikit-Learn framework was used on datasets with 1.200 rows and 168 environmental features, plus an additional column with the

Embrapa Uva e Vinho
Rua Livramento, nº 515
Caixa Postal 130
95701-008 Bento Gonçalves, RS
www.embrapa.br/uva-e-vinho
www.embrapa.br/fale-conosco/sac

Comitê Local de Publicações
Presidente
Henrique Pessoa dos Santos
Secretária-executiva
Renata Gava
Membros
Fernando José Hawerroth, Mauro Celso Zanus, Joelsio José Lazzarotto, Jorge Tonietto, Thor Vinícius Martins Fajardo, Alessandra Russi, Edgardo Aquiles Prado Perez, Fábio Ribeiro dos Santos, Luciana Elena Mendonça Prado, Michele Belas Coutinho Pereira e *Rochelle Martins Alvorcem*

Revisão de texto
Renata Gava
Normalização bibliográfica
Rochelle Martins Alvorcem
(CRB-10/1810)
Projeto gráfico
Leandro Sousa Fazio

Publicação digital: PDF

alerts to be classified. The workflow included: data processing and definition of features and target variable, normalization of features and numerical conversion of target variables, splitting data into training and testing sets, training individual models and combining them into a voting classifier, pipeline configuration and evaluation with cross-validation, hyperparameter optimization and evaluation of the optimized model on the test set and predictions with new features. Although the combined model achieved an accuracy of 97%, predictions with new data were lower (84%), showing slight overestimation. Nevertheless, the model presents itself as an effective candidate to be trained with field observation data in validation work for more accurate versions of Embrapa/MAHM.

Index terms: machine learning, support vector machines, logistic regression, random forest, data science.

Introdução

Os modelos de classificação em aprendizado de máquina (ML) (em inglês, *Machine Learning*) emergiram como ferramentas poderosas na agricultura moderna e vêm proporcionando avanços na gestão de pragas e doenças (Kumari et al., 2023). Analistas e técnicos cada vez mais vem incorporando o uso de algoritmos sofisticados como *Random Forest*, *Support Vector Machines* (SVM) e Redes Neurais/ *Deep Learning* dentre outros no manejo de pragas e doenças (Om et al., 2024). Atualmente, esses modelos analisam grandes volumes de dados, principalmente provenientes de imagens multiespectrais e sensores, permitindo a detecção precoce de problemas fitossanitários (Kukadiya e Meva, 2023). Ferramentas baseadas em ML facilitam a tomada de decisões precisas, promovendo práticas agrícolas mais eficientes e sustentáveis (Shaikh et al., 2022). A aplicação desses modelos não só aumenta a produtividade, mas também reduz a necessidade de intervenções químicas, contribuindo para uma agricultura mais sustentável (Cavalcanti, 2021; Patel; Suthar, 2022; Cavalcanti et al., 2024).

Há um número crescente de trabalhos que usam grandes massas de dados principalmente de imagens multiespectrais (Sethy et al., 2020). No entanto, há dados de sensores diversos para impulsionar modelos de aprendizado. Por exemplo, um estudo demonstrou que o algoritmo de máxima verossimilhança alcançou uma precisão de 88,10% na identificação de laranjeiras doentes utilizando imagens de alta resolução capturadas por drones, comparado a outras técnicas como *Support Vector Machine* (77,38%), *Spectral Angle Mapper* (76,19%), *K-Nearest Neighbor* (64,68%) e *Random Forest* (61,90%) (Díaz Rivera et al., 2024). Outro estudo aplicou técnicas de ML a dados multiespectrais para monitorar castanheiros, obtendo uma taxa de acurácia entre 86 e 91% na detecção de problemas fitossanitários e taxas entre 80 e 85% na identificação específica dos problemas (Padua et al., 2020). Além disso, a aplicação de modelos como *Support Vector Machines* em conjunto com redes neurais mostrou-se eficaz na classificação de doenças específicas em culturas como o arroz, com o modelo *ShuffleNet* combinado com SVM alcançando uma precisão de 89,37%, sensibilidade de 89,37%, especificidade de 94,68% e um tempo de computação muito pequeno (Sethy et al., 2020). Esses avanços não só devem possibilitar o aumento da produtividade agrícola, mas também promovem práticas mais sustentáveis e ecológicas. A integração de ML na agricultura é, portanto, uma abordagem promissora para enfrentar os desafios fitossanitários contemporâneos.

Em videira, os modelos de classificação estão transformando a viticultura, especialmente na identificação e gerenciamento de vinhedos por imagens. Por exemplo, um estudo desenvolveu modelos de ML para prever fenologia e risco de pragas em vinhedos, reduzindo a necessidade de tratamentos fitossanitários e melhorando a sustentabilidade da videira (Lacueva Pérez et al., 2020). Outro estudo utilizou imagens aéreas de alta precisão de vinhedos, melhorando a acurácia de detecção de objetos agrícolas de 89,6 para 94,27% (Treboux; Genoud, 2018). Além disso, a aplicação de *Deep Learning* para a classificação de imagens de folhas de videira alcançou uma precisão de 95,13% com um modelo CNN-GLS (rede neural convolucional associada a um método de quadrados mínimos generalizados), demonstrando alta capacidade de generalização para a identificação taxonômica de plantas (Kunduracioglu; Pacal, 2024). A integração de ML na viticultura é uma abordagem promissora para enfrentar os desafios fitossanitários contemporâneos.

Desde 2019, a Embrapa Uva e Vinho vem desenvolvendo um algoritmo de geração de alertas contra o míldio, baseado em dados provenientes de sensores de temperatura e umidade do ar instaladas em redes de estações meteorológicas de IoT (*Internet of Things*, em inglês, ou Internet das Coisas). Esses dados são usados para a composição de mapas temáticos (*heat maps*) de favorabilidade e alertas de aplicação de fungicida no instante e no posicionamento (georreferenciado) adequado por esse algoritmo: o Módulo de Alerta Georreferenciado de

Doença de Planta por *Heat Map* (Embrapa/MAHM). O aplicativo foi registrado no Instituto Nacional de Propriedade Intelectual (Inpi) com o Certificado n° BR512019002684-5, 2019 (Cavalcanti, 2021). A questão do georreferenciamento do alerta de pulverização é uma característica de inovação trazida por essa tecnologia e, até por isso, suscita uma enorme prudência nos ensaios de validação em condições de vinhedos. Até o momento, há ensaios de validação dos alertas do MAHM em três safras: 2019/2020 (Cavalcanti, 2021), 2020/2021 e 2022/2023 (Cavalcanti et al., 2024), todos bem sucedidos em termos de controle do míldio, doença causada pelo patógeno *Plasmopara viticola*. No entanto, os trabalhos de validação foram conduzidos basicamente para avaliar o *timing* (freqüência) de aplicação, uma vez que para avaliação do georreferenciamento existe a necessidade de um modelo de aprendizagem de máquina capaz de incorporar novas informações trazidas de inspeções ao vinhedo, ou seja, da realidade. Essas informações devem ser basicamente associadas à informação de entrada de infecção e/ou de epidemias de míldio em quadrantes descobertos por alertas anteriores que prescindiram de pulverização, e promovendo uma alteração "manual" e uma reprogramação nos alertas. Para isso, um modelo de aprendizado e classificação seria capaz de promover uma ferramenta ativa na calibração da parte do algoritmo responsável pelo gerenciamento dos alertas (o método "GMAHM"), o que produziria um refinamento e aumento da acurácia no georreferenciamento dos alertas, possibilitando o desenvolvimento de novas versões não só do método, mas como de todo algoritmo.

Por esse motivo, o objetivo deste trabalho foi desenvolver um modelo de aprendizado supervisionado em ML para classificação de padrões gerados pelo GMAHM e, a partir da validação desse modelo, obter uma ferramenta capaz de "aprender" nos padrões reais (em condições de campo) de distribuição espacial de míldio, no caso de infecção e/ou surtos da doença em pontos não cobertos pelos alertas anteriores. O modelo supervisionado de classificação foi construído por meio do f*ramework Scikit-learn* que consiste num módulo em *Python* para aprendizado de máquina, desenvolvido sobre a biblioteca *SciPy*, que fornece ferramentas simples e eficientes para análise e modelagem de dados. Ele inclui uma variedade de algoritmos de aprendizado supervisionado, semi e não supervisionado, bem como ferramentas para pré-processamento, seleção de modelos e avaliação de desempenho.

Material e métodos

Fonte dos dados

Foram construídos dois conjuntos de dados (*datasets*) básicos: 1) para uso no desenvolvimento do modelo ('df_treino'/'df_teste') a partir de três saídas de alertas do MAHM3-O2 (MAHM) (Cavalcanti et al., 2024) sem cadastro de vinhedos e abordando uma das microrregiões estudadas na safra 2022/2023 por Cavalcanti et al. (2024); e 2) dados "novos" para gerar uma predição, ao final do desenvolvimento do modelo. Ambos os *datasets* usados tinham uma estrutura 1.200 linhas x 168 colunas de dados mais uma última coluna (169) de classificação. Os dados dos *datasets* foram organizados obedecendo à seguinte estrutura: 1) linhas – dados relacionados a cada um dos diferentes 400 pixels (quadrantes) gerados pela simulação do algoritmo em uma resolução 20 x 20; 2) colunas – dados relacionados à favorabilidade calculada para as 168 horas de cobertura semanal, sobre dados de simulação. As 168 primeiras colunas compostas de leituras de favorabilidade [dados do tipo float64(168)] foram consideradas a serem usadas como "características" ou "atributos" (*features*). Na coluna 169 foram dispostos os alertas de simulação atribuídos a cada quadrante (cada linha), e essa informação foi considerada como "variável resposta" ao qual o estudo de modelagem vai se propor a predizer (*target*). O critério dos alertas seguem as cores: vermelho, significando "pulverização imediata"; amarelo, significando "aguardar uma semana"; a continuar amarelo, "pulverizar imediatamente"; e verde, significando "não pulverizar" (Cavalcanti, 2021).

Organização do *framework* e incorporação dos dados

Em um notebook .ipynb foram importadas as bibliotecas *numpy* e pandas para manipulação de matrizes (*arrays*) e estruturas de dados. Os dados foram carregados a partir de um arquivo .csv e visualizados. Informações gerais sobre o *dataframe* (df) de dados foram obtidas, incluindo o número de entradas, o tipo de dados de cada coluna e a quantidade de valores não nulos. Foi realizada a contagem dos valores na coluna "alertas", incluindo valores nulos (NaN), sem descartar nenhum valor. Por fim, foram geradas estatísticas descritivas resumidas.

Preparação dos dados: *features* e *target*

Os dados foram filtrados para remover quaisquer entradas com valores nulos, resultando no df "dados_rotulados". As características (*features*)

foram definidas removendo-se a coluna “alertas” do df “dados_rotulados”, criando “X” com as variáveis explicativas. A variável alvo (*target*) foi definida como a coluna “alertas” do *DataFrame* “dados_rotulados”, armazenada na série “y”. Em seguida, foi realizada uma contagem dos valores na coluna “alertas” para entender a distribuição das categorias. O “*LabelEncoder*” da biblioteca “sklearn.preprocessing” foi importado e utilizado para transformar os rótulos da variável alvo “y” em valores numéricos. Finalmente, os primeiros 300 valores transformados da variável “y” foram impressos para verificação.

Normalização e divisão dos dados

Foram importados os escaladores *MinMaxScaler* e *StandardScaler* da biblioteca sklearn.preprocessing, embora apenas o *StandardScaler* tenha sido utilizado neste caso, pois ele proporcionou melhores resultados no classification_report (y_teste, y_previsto). Os dados foram normalizados usando o StandardScaler, e o resultado foi armazenado no DataFrame “X_normalizado”, mantendo as mesmas colunas de “X”. Em seguida, foram importados “train_test_split”, “GridSearchCV” e “cross_validate” da biblioteca “sklearn.model_selection”. Os dados normalizados “X_normalizado” e os rótulos “y” foram divididos em conjuntos de treino e teste usando “train_test_split”, com estratificação pela variável alvo “y” e uma semente aleatória (“random_state”) definida como 10 para garantir a reprodutibilidade nesse caso.

Modelos de aprendizado supervisionado usados para classificação e combinação

Foi utilizado o modelo SVC (*Support Vector Machine Classifier*) com um *kernel* (função de transformação) linear para treinar e ajustar os dados de treino, com o objetivo de realizar a classificação. Após ajustar o modelo, as predições foram feitas no conjunto de teste e avaliadas utilizando o relatório de classificação (“classification_report”), que fornece métricas como precisão, *recall* e F1-score. Além disso, uma matriz de confusão foi gerada e visualizada para entender melhor a performance do modelo nas diferentes classes. Da mesma forma, foi treinado um modelo de Regressão Logística (“*LogisticRegression*”) configurado para classificação multiclasse, e suas predições e desempenho foram avaliados de maneira similar. Por fim, foi utilizado o “*RandomForestClassifier*” para treinar e ajustar os dados de treino, com o desempenho avaliado por meio de predições no conjunto de teste e geração do relatório de classificação. Esses passos foram realizados para comparar a eficácia de diferentes modelos de classificação.

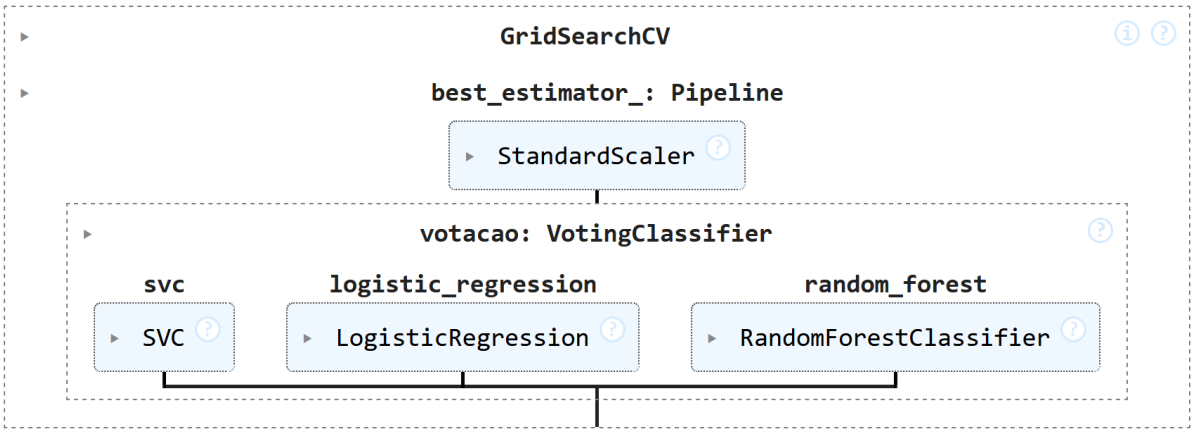

Figura 1. Híperparâmetros otimizados pelo método “GridSearchCV”, por pesquisa exaustiva em uma biblioteca de opções para maximizar a performance do modelo. O *pipeline* inicia com a normalização dos dados, seguido pela combinação de três classificadores: SVC (SVM), Regressão Logística e *Random Forest*. A estrutura equidistante da grade reflete a busca simultânea por ajustes ótimos em cada modelo individual. A ligeira tendência para Regressão Logística sugere uma maior participação desse classificador no equilíbrio do modelo.

Pipeline para combinação de três classificadores e validação cruzada

Foi criado um modelo "VotingClassifier", que combinou os três estimadores selecionados em 2.5.: SVC, Regressão Logística e *Random Forest*, com o objetivo de utilizar a votação suave ("voting = soft") para melhorar a robustez das predições, combinando as probabilidades preditivas dos três modelos (Figura 1). Este modelo de votação foi então integrado a um *pipeline* que começa com a normalização dos dados, garantindo que todas as características (*features*) estejam na mesma escala antes de serem alimentadas nos modelos de classificação. Para avaliar o desempenho do *pipeline* completo, foi utilizada a função *cross_validate* com validação cruzada de cinco divisões de dados (cv = 5), o que permite medir a acurácia de treino e teste em diferentes subconjuntos dos dados de treino. As médias das acurácias de treino e teste foram calculadas e impressas para verificar a consistência do modelo e para identificar possíveis sinais de *overfitting* ou *underfitting*.

Otimização dos três modelos para o modelo de combinação

Foi definida uma grade de parâmetros para otimizar os hiperparâmetros do modelo "V*otingClassifier*" combinado, com configurações específicas para os três estimadores (classificadores) individuais. Para o SVC, os parâmetros "C"e *kernel* foram ajustados; para a Regressão Logística, os parâmetros "C" e "solver" foram ajustados; e para o *Random Forest*, os parâmetros "n_estimators" e "max_depth" foram ajustados. Em seguida, foi utilizada a função "GridSearchCV" para realizar a busca em grade (*grid search*) com validação cruzada de cinco *folds* ("cv = 5"), visando encontrar a melhor combinação de hiperparâmetros que maximizam a performance do *pipeline*. Após ajustar o modelo com os dados de treino ("X_treino" e "y_treino"), foram verificados os melhores estimadores e parâmetros encontrados pelo "GridSearchCV". O melhor modelo combinado ("mod_combinado") foi selecionado e a melhor acurácia obtida na validação cruzada foi impressa, juntamente com os melhores parâmetros. Finalmente, o modelo ajustado foi avaliado no conjunto de teste ("X_teste" e "y_teste"), e a acurácia das predições no conjunto de teste foi calculada e impressa utilizando a métrica "accuracy_score".

Resultados e discussão

O objetivo do trabalho consistiu na proposta de um modelo de aprendizado supervisionado em *Machine Learning* (ML) capaz de compreender a estrutura numérica do método gerador de alertas do MAHM, o "GMAHM", com o *framework Scikit-Learn*. Esse modelo supervisionado deveria ser capaz de apreender os dados de "features" e "target" e montar a modelagem numérica do GMAHM para o aprendizado supervisionado e utilização da modelagem em dados novos, provenientes de inspeções a campo, capaz de gerar predições ainda mais acuradas. O modelo proposto deve ser principalmente usado em trabalhos de validação e calibração após verificação de falhas de cobertura de alerta (pontos e/ou surtos de míldio) pela versão atual (ou anterior) do GMAHM (Cavalcanti, 2021; Cavalcanti et al., 2024). Basicamente, o trabalho propõe uma nova versão do método GMAHM capaz de ser treinada com dados obtidos por informações externas, evidências de míldio em quadrantes com falha de proteção, e aprender (incorporar) os novos padrões.

Antes deste estudo, o GMAHM apresentava meramente um gerenciamento baseado em "números de corte" livremente atribuídos sobre percentuais de picos de favorabilidade (ao míldio) acumulada a cada semana (Figura 2). Essa versão inicial foi validada por três safras e foi eficaz em gerar alertas capazes de proteger vinhedos experimentais de *Vitis vinifera* e *Vitis labrusca* (Cavalcanti et al., 2024). No entanto, essa versão "mecanística" não dispunha de uma abordagem racional nem para incorporar dados externos, tampouco para efetuar correções na estrutura da geração dos mapas de alerta (*heat maps*) em caso de falhas de cobertura no georreferenciamento do algoritmo.

Para isso, foi avaliada uma estratégica de combinação de três modelos de classificação para um conjunto de dados com muitas entradas de dados numéricos contínuos (alta dimensionalidade) e uma coluna categórica com três níveis (os três alertas trabalhados). Foram optados pela combinação de três modelos dos inúmeros disponibilizados pelo framework: a) o *Random Forest Classifier*, capaz de lidar bem com dados de alta dimensionalidade e estimativas de importância das características (Breiman, 2001); b) o *Support Vector Machine* (SVM), também adequado à condições de alta dimensionalidade, com diferentes *kernels* para ajustes de complexidade (Cortes; Vapnik, 1995); e c) Regressão Logística, por sua simplicidade e robustez (Cox, 1958).

```
                aafa[c2] = aafa[c2] + ((((fauq[c2, (p - 1)] * fatq[c2, (p - 1)]) +
(fauq[c2, p] * fatq[c2, p])) / 2) * (p - (p - 1)))
            ## 'Estimador' do Ozzy  ** PONTO DE CALIBRACAO
            fa[c2] = 1.0 * aafa[c2] * fos[c2]  ### ********* CAF MODIF PARA O BPD 36
            ## Atribui a cor da estimativa de alerta a cada quadrante  ** PONTO DE
CALIBRACAO
            tre = 0.65 * np.exp(0.464 * sebl) # 6.61 e o valor quando SEBL = 5 (default)
            ## GMAHM  ** PONTO DE CALIBRACAO
            if(fa[c2] > (690130 * (6.61 / tre))):
                cor = 'red'
            if(fa[c2] > (662035 * (6.61 / tre)) and fa[c2] <= (690130 * (6.61 / tre))):
                cor = 'yellow'
            if(fa[c2] <= (662035 * (6.61 / tre))):
                cor = 'green'
            ## Plota
            x0 = qx[i - 1]
            y0 = (qy[(res)] - qy[f - 1])
            deltax = qx[1] - qx[0]
            deltay = qy[1] - qy[0]
            quadr = plt.Rectangle((x0,y0), width=deltax, height=-deltay, facecolor=cor,
edgecolor='grey')
            ax.add_patch(quadr)
            print(x0, y0, deltax, deltay) # check!
    return
```

Figura 2. Segmento de *script* em *Python* do simulador do MAHM3-O2 contendo o método GMAHM, que gerencia as cores como uma função interna a *loops* que congregam a dinâmica de situações ambientais e os n-quadrantes, na formação do mapa temático. *Box* vermelho detalha ao critério de escolha "mecanístico" atribuído à decisão dos alertas, na primeira versão.

A abordagem de combinação de modelos seguiu uma sequência típica de: a) carregamento e tratamento dos dados; b) as características (*features*) e a variável alvo (*target*, Figura 3) foram separadas em "X" e "y"; c) as características foram normalizadas e as variáveis categóricas alvo sofreram conversão numérica; d) os dados normalizados foram divididos em conjuntos de treino e teste utilizando "*train_test_split*"; e) os modelos individuais optados foram treinados nos dados de treino; f) os modelos treinados foram combinados em um "*VotingClassifier*"; g) um *pipeline* foi configurado para incluir a normalização e a votação; h) o *pipeline* foi avaliado utilizando validação cruzada; i) "GridSearchCV" foi executado para encontrar a melhor combinação de hiperparâmetros (otimização); j) o modelo otimizado foi avaliado no conjunto de teste e as métricas foram avaliadas; e l) foram feitas predições com novas *features*, com o modelo combinado final.

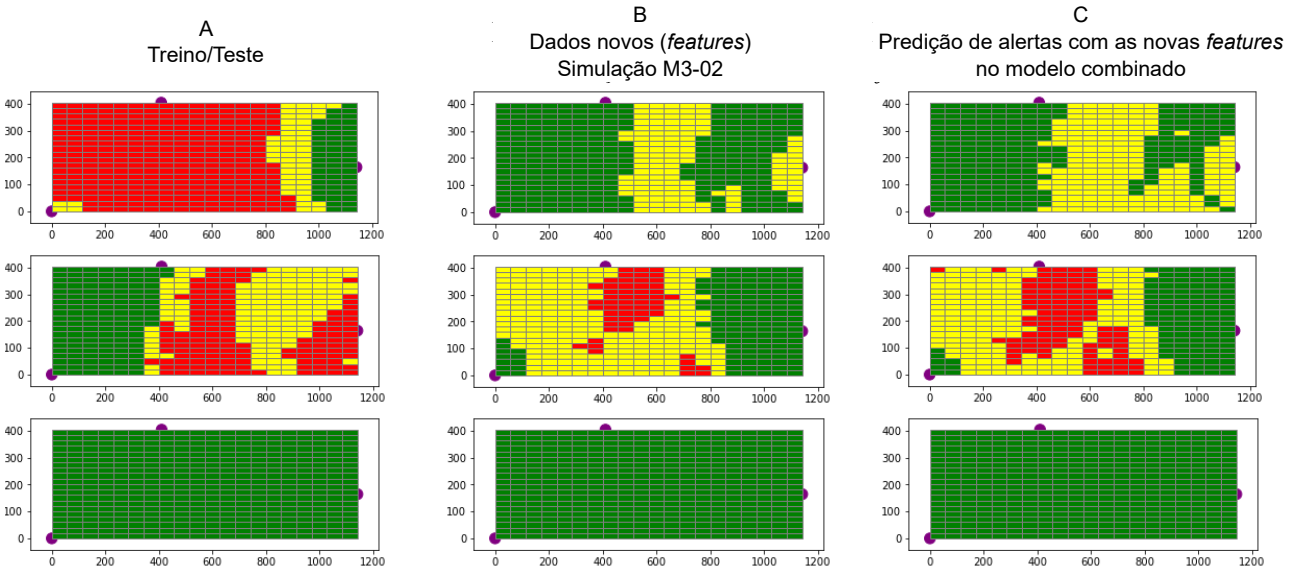

Figura 3. Mapas dos alertas associados aos conjuntos de dados: (A) "treino" e "teste" obtidos por simulação MAHM3-O2 (M3-O2) para composição do modelo supervisionado; (B) simulação MAHM3-O2 gerando *features* de novos dados para predições de classificação; e (C) mapas dos alertas preditos pelo modelo combinado, no *Scikit-learn*.

Rodando o modelo, as métricas obtidas indicaram que a abordagem combinada de classificação apresentou um desempenho excelente, com uma acurácia geral de 0,97, significando que 97% das predições foram corretas. A precisão, *recall* (revocação/sensibilidade) e f1-score são altos para todas as classes, com a classe 0 (alerta verde) apresentando uma precisão de 0,98 e um *recall* perfeito de 1,00, resultando em um f1-score de 0,99. A classe 1 (alerta amarelo) teve uma precisão de 0,95, *recall* de 0,94 e f1-score de 0,95, enquanto a classe 2 (alerta vermelho) apresentou uma precisão de 0,95, *recall* de 0,93 e f1-score de 0,94. As médias macro e ponderada das métricas foram ambas de 0,96 e 0,97, respectivamente, refletindo que o modelo é eficaz tanto em situações de classes equilibradas quanto desbalanceadas. A classe 0 teve um desempenho ligeiramente superior, possivelmente devido às características mais distintivas ou um leve desbalanceamento nos dados. No geral, as métricas indicam que o modelo é robusto, generaliza bem e trata as classes de forma equitativa, sugerindo uma excelente capacidade de predição. O desempenho do modelo combinado é reflexo dos ajustes com os três modelos optados para o estudo que também ajustaram muito bem, como pode ser verificado nas matrizes de confusão (Figura 4).

A combinação de hiperparâmetros que otimizam o desempenho do modelo "*VotingClassifier*" foi composto pelos três classificadores (Figura 1). Os melhores hiperparâmetros encontrados pelo "GridSearchCV" foram: para a Regressão Logística, "C = 10", indicando menor regularização, e "solver = saga", adequado para grandes conjuntos de dados; para o *Random Forest*, "max_depth = None", permitindo que as árvores cresçam sem restrições de profundidade, e "n_estimators = 50", proporcionando um equilíbrio entre variância e viés; e para o SVC, "C = 10", indicando menor regularização, e kernel = "linear", utilizando um hiperplano linear para a separação das classes (alertas). Essa combinação de parâmetros foi determinada como a melhor configuração para maximizar a performance do modelo combinado nos dados fornecidos.

As métricas de precisão, *recall*, f1-score e suporte (*support*) fornecidas por uma instrução "classification_report()" são essenciais para avaliar o desempenho de um modelo de classificação. A precisão indica a proporção de predições corretas para uma classe específica entre todas as predições feitas para essa classe, refletindo a exatidão do modelo em identificar exemplos positivos sem incluir falsos positivos (Manning et al., 2009). O *recall* (sensibilidade ou "revocação") mede a proporção de exemplos positivos corretamente identificados pelo modelo entre todos os exemplos que realmente pertencem àquela classe, sendo crucial para avaliar a capacidade do modelo em detectar todos os casos positivos, minimizando falsos negativos (Van Rijsbergen, 1979). O f1-score (métrica, medida, índice f1) é a média harmônica da precisão e do *recall*, oferecendo um balanço entre essas duas métricas, especialmente útil em cenários de classes desbalanceadas, pois uma alta precisão combinada com um *recall* baixo, ou vice-versa, indicaria um desempenho desigual que o f1-score captura melhor (Sasaki, 2007).

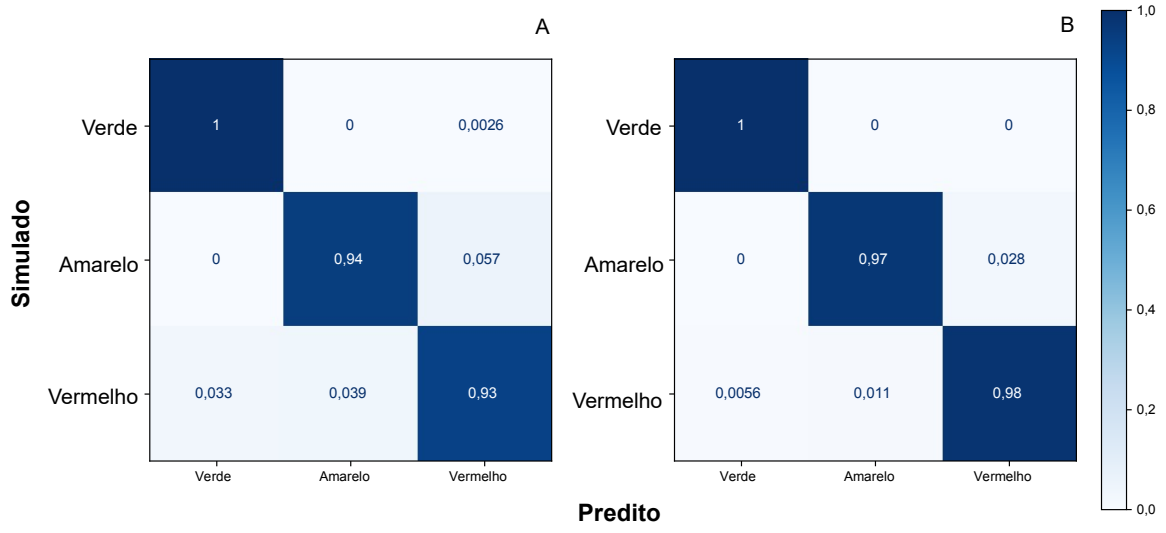

Figura 4. (A) Matriz de confusão associada ao modelo SVM [svm.fit(X_treino, y_treino)]; (B) matriz de confusão associada ao modelo combinado (SVC, Regressão Linear e *Random Forest*) submetido à normalização, classificação por votação, validação cruzada e otimização de hiperparâmetros.

O suporte (no "classification_report") representa o número de exemplos reais em cada classe, servindo como base para calcular as outras métricas e permitindo entender a distribuição das classes nos dados. Juntas, essas métricas permitem uma compreensão abrangente do desempenho do modelo, onde a precisão e o *recall* individualmente informam sobre a taxa de acertos e a cobertura de predições corretas, respectivamente, enquanto o f1-score sintetiza essas informações em um único valor equilibrado. A análise dessas métricas para cada classe e suas médias ponderada e macro fornecem *insights* sobre a robustez e a equidade do modelo na classificação de todas as classes presentes nos dados.

Ao final da montagem do modelo combinado, foram rodadas predições sobre dados novos de simulação em outra situação completamente distinta de dinâmica microclimática em função das três microestações de IoT, com várias semanas de distância entre os conjuntos de dados. Em várias tentativas as predições demonstraram uma leve superestimação, em jargão de ML, *overfitting*. A queda na acurácia do modelo desceu para aproximadamente 0,84 (84%), que ainda é considerado um patamar eficaz. No entanto, para melhorar as predições, uma abordagem seria revisar e ajustar a complexidade do modelo, possivelmente reduzindo a complexidade dos classificadores (por exemplo, ajustando os parâmetros "max_depth" do *Random Forest* ou "C" do SVC), e ajustar. Além disso, é fundamental garantir que os dados novos estejam pré-processados e normalizados de maneira consistente com os dados de treinamento.

No presente trabalho, dois operadores de normalização foram experimentados, no entanto sem redução na superestimação. Considera-se, nos próximos avanços, a expansão do conjunto de dados de treinamento com mais exemplos representativos das situações encontradas nos dados novos, bem como utilização de técnicas de validação cruzada mais robustas, com mais subconjunto de dados (k > 5), para avaliar a generalização do modelo. Avaliar e ajustar a seleção de *features* pode também ajudar a eliminar redundâncias e ruídos, melhorando a capacidade de generalização do modelo.

Conclusões

1) Um modelo de aprendizado supervisionado de máquina foi desenvolvido como candidato a substituir o método original para geração de alertas do MAHM ("GMAHM"). Uma abordagem combinando três classificadores (SVM, Regressão Logística e *Random Forest*) foi experimentada sobre um conjunto de dados de treinamento e teste, com acurária de 97%.
2) Quando submetido a dados novos para classificação, a precisão do modelo se reduz para 84%, superestimando predições associadas a alguns quadrantes.
3) No entanto, como uma primeira versão, considera-se que o modelo ajustou satisfatoriamente reproduzindo o mapa temático.
4) Há perspectivas para melhorias sobre o modelo proposto, com: a) aprofundamento de técnicas para prevenção de *overfitting* (existem inúmeras); b) explorar diferentes tratamentos de características (*features*) para melhorar a qualidade das predições em dados novos; c) experimentar outros classificadores; d) incrementar a validação cruzada, com outros subconjuntos de dados e divisões, para melhorar a generalização; e e) testar outras técnicas de pré-processamento e normalização de dados para reforçar a consistência das predições em dados novos.

Referências

BREIMAN, L. Random Forests. **Machine Learning**. v. 45, n. 1, p. 5-32, 2001. DOI: 10.1023/ A:1010933404324.

CAVALCANTI, F. R. **Algoritmo (MAHM) para alerta georreferenciado de doença em redes de sensoriamento IoT de microclima**: calibração e teste de um método para míldio, em dois vinhedos. Bento Gonçalves, RS: Embrapa Uva e Vinho, ago. 2021. (Embrapa Uva e Vinho. Circular técnica 163). 25 p. Disponível em: http://www.infoteca.cnptia.embrapa.br/infoteca/handle/doc/1134000. Acesso em: 8 ago. 2024.

CAVALCANTI, F. R; BOTTON, M.; FIORAVANÇO, J. C. **Atualizações de alertas para controle do míldio da videira com o Algoritmo Embrapa/MAHM**. Bento Gonçalves, RS: Embrapa Uva e Vinho, jul. 2024. (Embrapa Uva e Vinho. Boletim de Pesquisa e Desenvolvimento 30). 25p. Disponível em: http://www.infoteca.cnptia.embrapa.br/infoteca/handle/doc/1165921. Acesso em: 2 set. 2024.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, v. 20, n. 3, p. 273-297, 1995. DOI: 10.1007/BF00994018.

COX, D. R. The Regression Analysis of Binary Sequences. **Journal of the Royal Statistical Society: Series B (Methodological)**, v. 20, n. 2, p. 215-232, 1958. DOI: 10.1111/j.2517-6161.1958.tb00292.x.

DÍAZ RIVERA, J. C.; AGUIRRE-SALADO, C. A.; LOREDO-OSTI, C.; ESCOTO-RODRIGUEZ, M. Identification of the phytosanitary status of trees using

machine learning and very high spatial resolution images. **Scientia Agropecuaria**, v. 15, n. 2, p. 177-189, Abril/Jun. 2024. DOI: https://dx.doi.org/10.17268/sci.agropecu.2024.013

KUKADIYA, H.; MEVA, D. Machine learning in agriculture for crop diseases identification: A Survey. **International Journal of Research - GRANTHAALAYAH**, v. 11, n. 3, p. 276-284, 2023. DOI: https://dx.doi.org/10.29121/granthaalayah.v11.i3.2023.5099.

KUMARI, S.; VENKATESH, V. G.; TAN, F. T. C.; BHARATHI, S. V.; RAMASUBRAMANIAN, M.; SHI, Y. Application of machine learning and artificial intelligence on agriculture supply chain: a comprehensive review and future research directions. **Annals of Operations Research**, online, sept. 2023. DOI: https://dx.doi.org/10.1007/s10479-023-05556-3.

LACUEVA PÉREZ, F. J.; HOYO-ALONSO, R. D.; BARRIUSO VARGAS, J. J.; ARTIGAS, S. I. Towards improving agriculture sustainability through multifactorial Machine Learning. **JJI3A: Jornada de Jóvenes Investigadores del I3A**, v. 8, 2020. DOI:10.26754/JJII3A.4868.

MANNING, C. D.; RAGHAVAN, P.; SCHUTZE, H. An introduction to information retrieval. **Cambridge University Press**, 2009. ISBN: 978-0521865715.

OM, G.; BILLA, S. R.; MALIK, V.; BHARATH, E.; SHARMA, S. Grapevine fruits disease detection using different deep learning models. **Multimedia Tools and Applications**, 2024. DOI: https://doi.org/10.1007/s11042-024-19036-8.

PADUA, L.; MARQUES, P.; MARTINS, L.; SOUSA, A.; PERES, E.; SOUSA, J. J. Monitoring of chestnut trees using Machine Learning techniques applied to UAV-Based multispectral data. **Remote Sensing**, v. 12, n. 18, p. 3032, Sept. 2020. DOI: https://dx.doi.org/10.3390/rs12183032.

PATEL, P. M.; SUTHAR, D. A. Agriculture crop enhancing identification and classification using Machine Learning techniques. **International Journal of Advanced Research in Computer and Communication Engineering**, v. 11, n. 4, p. 103-106, April 2022. DOI: https://dx.doi.org/10.17148/ijarcce.2022.114103.

SASAKI, Y. **The truth of the F-measure**. Technical report, School of Computer Science, University of Manchester, 2007.

SETHY, P. K.; BARPANDA, N. K.; RATH, A. K.; RAJPOOTt, S. C. Rice (Oryza sativa) panicle blast grading using support vector machine based on deep features of small CNN. **Archives of Phytopatology and Plant Protection**, v. 54, p. 15-16, Dec. 2020. DOI:10.1080/03235408.2020.1869386.

SHAIKH, T. A.; RASOOL, T.; LONE, F. R. Towards leveraging the role of machine learning and artificial intelligence in precision agriculture and smart farming. **Computers and Electronics in Agriculture**, v. 198, p. 107119, July 2022. DOI: https://dx.doi.org/ 10.1016/j.compag.2022.107119.

KUNDURACIOGLU, I.; PACAL, I. Advancements in deep learning for accurate classification of grape leaves and diagnosis of grape diseases. **Journal of Plant Diseases and Protection**, v. 131, p. 1061–1080, MArch 2024. DOI: https://doi.org/10.1007/s41348-024-00896-z.

TREBOUX, J.; GENOUD, D. Improved Machine Learning methodology for high precision agriculture. In: Global Internet of Things Summit (GIoTS), 2018, Bilbao, Spain. **Abstracts** [...].,Bilbao: IEEE, 15 Nov. 2018. DOI: 10.1109/GIOTS.2018.8534558. VAN RIJSBERGEN, C. J. **Information Retrieval**. Butterworth-Heinemann, 1979. ISBN: 978-0408709295.

Embrapa
Ministério da Agricultura e Pecuária

CGPE 19049