Avaliação de técnicas de balanceamento na classificação de aceitabilidade de carros

Lucas Ferreira Paiva[1,2], Allan Fernando Oliveira de Mattos[1,2], Lucas de Assis Silva[1], Juscimara Gomes Avelino[2], George Darmiton da Cunha Cavalcanti[2]

[1]Embraer S.A.

[2]Centro de Informática – Universidade Federal de Pernambuco (UFPE)

{{lucas.paiva, allan.mattos, lucas.assis}@embraer.com.br,

{jga2,gdcc}@cin.ufpe.br}

Abstract. *Car acceptability involves classifying a vehicle based on its physical and financial characteristics. This type of analysis assists in the decision of acquiring or not a specific car model. In this study, the objective was to evaluate the impact of using undersampling, oversampling, and a combination of both techniques on eight machine learning models. For each balancing technique and model, hyperparameter optimization and attribute selection were applied. The results obtained in this study surpassed the state-of-the-art for SVM. Furthermore, it was possible to observe the improvement of simpler models with the use of balancing techniques.*

Resumo. *A aceitabilidade de carros consiste em classificar um veículo com base nas suas características físicas e financeiras. Esse tipo de análise auxilia na aquisição, ou não, de um determinado modelo de automóvel. Neste estudo, o objetivo foi avaliar o impacto do uso de técnicas de subamostragem, sobreamostragem e uma combinação das duas técnicas em oito modelos de aprendizado de máquina. Para cada técnica de balanceamento e modelo foi utilizado otimização de hiper-hiperparâmetros e seleção de atributos. Os resultados obtidos neste estudo superaram o estado da arte para o SVM. Além disso, foi possível notar a melhora de modelos mais simples com o uso das técnicas de balanceamento.*

1. Introdução

Os algoritmos de aprendizado de máquina supervisionados são capazes de aprender a partir de exemplos por meio do reconhecimento de padrão e a partir disso, classificar novos dados apresentados. Esta funcionalidade permite a utilização desses algoritmos em uma gama variada de aplicações, como detecção de doença em plantas [Shruthi et al. 2019], classificação de animais [Lopez-Vazquez et al. 2020], auxílio ao diagnóstico médico em exames de imagem [Maier et al. 2019], avaliação de crédito [Yu et al. 2022] e estimação de ritmo de músicas [Ferreira-Paiva et al. 2022].

Quando se trata em adquirir um carro, diversos fatores devem ser considerados na avaliação do veículo, cada um com um grau de importância diferente na tomada de decisão. Por exemplo, o veículo que possuir maior preço não necessariamente oferece os melhores hiperparâmetros para a decisão de compra, uma vez que o modelo e fabricante

impactam em outros fatores como segurança e custo de manutenção. Visando fomentar essa discussão, a Universidade da Califórnia Irvine disponibiliza em seu repositório um conjunto de dados dedicado à classificação de aceitabilidade de carros. Os dados contêm os seguintes fatores associados à classificação de aceitabilidade de carros: custo da manutenção, número de passageiros, número de portas, tamanho do porta-malas e segurança.

Devido à importância do tema e a disponibilidade de dados, foram encontrados diversos estudos na área de aprendizado de máquina e reconhecimento de padrões que exploraram esse conjunto de dados [Makki et al. 2011, Awwalu et al. 2014, Potdar et al. 2017, Rehman et al. 2018, Uzut and Buyrukoğlu 2020, Jain and Vishwakarma 2017]. Apesar de se tratar de um conjunto de dados desbalanceado, nenhum dos trabalhos encontrados utilizaram técnicas de balanceamento durante o pré-processamento. Além disso, somente um trabalho utilizou métricas de avaliação apropriadas para dados desbalanceados [Jain and Vishwakarma 2017]. Os demais trabalhos usaram somente a acurácia para avaliar os modelos, o que é reconhecidamente inadequado por se tratar de um banco de dados desbalanceado [De Diego et al. 2022].

Em decorrência da escassez de abordagens que consideram o desbalanceamento do conjunto de dados, este trabalho tem como objetivo avaliar o desempenho de diferentes técnicas de balanceamento de dados na classificação de aceitabilidade de carros com modelos de aprendizado de máquina. Além disso, para cada modelo e técnica de balanceamento foi realizado otimização de hiperparâmetros e seleção de atributos.

Neste estudo, o SVM alcançou uma acurácia superior ao estado da arte, mesmo sem a utilização de técnicas de balanceamento. Como principal contribuição foi observado a eficácia da combinação de seleção de atributos e *oversampling* para melhorar o desempenho dos modelos com baixo desempenho sem o uso balanceamento. Além disso, a importância de utilizar métricas abrangentes, especialmente aquelas que avaliam o desempenho por classe, é ressaltada como uma contribuição para avaliações mais completas dos modelos.

As seções seguintes do texto discutem técnicas de pré-processamento, incluindo balanceamento de instâncias e seleção de atributos, seguidas da descrição do banco de dados utilizado. Em seguida, são apresentados os experimentos realizados, que envolvem o balanceamento, seleção de hiperparâmetros e avaliação do desempenho dos modelos. Os resultados obtidos são abordados, incluindo a comparação dos modelos sem pré-processamento, o efeito das técnicas utilizadas, o desempenho dos melhores modelos por *fold* e a comparação com a literatura. Por fim, as conclusões são apresentadas.

2. Técnicas de pré-processamento

Nesta seção, duas estratégias de pré-processamento são apresentadas: balanceamento de instâncias e seleção de atributos.

2.1. Balanceamento de instâncias

O balanceamento de instâncias é uma técnica com finalidade de aumentar ou reduzir o número de instâncias do conjunto de dados. Três tipos de balanceamento, são comuns na literatura: subamostragem, sobreamostragem e híbrido [Krawczyk 2016].

A subamostragem, também conhecida como *undersampling*, consiste em reduzir o número de instâncias das classes majoritárias, isto é, as classes com mais instâncias, até que tenham aproximadamente ou a mesma quantidade de instâncias da classe minoritária. Neste estudo, foi utilizado o algoritmo "*random under sampler*" que realiza a redução populacional das classes através da remoção aleatória de instâncias do conjunto de dados até que todas classes do conjunto tenham o mesmo tamanho da classe minoritária.

A sobreamostragem, ou *oversampling*, é uma técnica de balanceamento de instâncias que, de forma resumida, realiza o aumento das instâncias mantendo o tamanho da classe majoritária e populando as classes minoritárias. Um algoritmo comum utilizado para realizar a sobreamostragem é o "*Synthetic Minority Over-sampling Technique* (SMOTE)". O algoritmo realiza a sobreamostragem da classe minoritária através da criação de exemplos sintéticos, pegando cada amostra e introduzindo exemplos sintéticos ao longo dos segmentos de linha que unem cada amostra a qualquer ou todos os vizinhos mais próximos da mesma classe [Chawla et al. 2002].

O método híbrido consiste na redução parcial do número de instâncias da classe majoritária e no aumento da quantidade de instâncias da classe minoritária. Para obter um balanceamento híbrido, é recomendado o uso combinado das técnicas de sub e sobreamostragem. Uma possibilidade é a redução parcial apenas da classe majoritária utilizando o *random under sampler* e o aumento das classes minoritárias até se igualarem à classe majoritária através do uso do SMOTE [Chawla et al. 2002].

O balanceamento de dados pode facilitar o aprendizado de algoritmos uma vez que equilibra a representatividade das classes inibindo que os algoritmos sejam ajustados a classificar sempre as classes majoritárias. Por outro lado, esse equilíbrio criado não reflete a natureza real dos dados. Para que isso não se torne um problema, é necessário que os dados utilizados para o teste do modelo não sejam balanceados [Krawczyk 2016].

2.2. Seleção de atributos

A seleção de atributos é uma importante técnica usada para escolher um subconjunto de atributos com maior relevância para o desenvolvimento de modelos de aprendizado de máquina [Guyon and Elisseeff 2003]. A utilização dessa técnica pode levar à redução do custo computacional, à melhoria da compreensão do problema modelado (características irrelevantes tendem a ser desconsideradas e interpretar modelos com menos atributos tende a ser mais fácil) e ao aumento do desempenho das predições (atributos irrelevantes podem causar, por exemplo, sobreajuste) [Chandrashekar and Sahin 2014].

Para selecionar os atributos, pode-se utilizar conhecimento de negócio [Guyon and Elisseeff 2003]; técnicas específicas, como, por exemplo: filtro, *embedded* e *wrapper*; ou até mesmo uma abordagem que una as diferentes técnicas, chamada de híbrida [Jović et al. 2015]. Assim, para escolher a abordagem a ser utilizada, pode-se balancear o custo computacional e a performance esperada. E, dentre esses métodos, o *wrapper* é conhecido por ser computacionalmente demandante, mas ele tende a prover melhores resultados [Chandrashekar and Sahin 2014].

Na seleção de um subconjunto de atributos com o *wrapper*, são comparadas as performances de modelos treinados com diferentes conjuntos de atributos. Assim, pode-se selecionar o conjunto de atributos do modelo que obteve o melhor desempenho. No entanto, como analisar todas combinações possíveis de atributos é computacionalmente

demandante, várias metodologias foram criadas para definir o subconjunto ideal, como a Seleção Sequencial de Atributos, que adiciona/remove um atributo em cada iteração. Neste trabalho, foi utilizado um método que segue essa metodologia, chamado *Sequential Backward Selection* (SBS), no qual é definido um conjunto inicial com vários atributos e, em cada iteração, um atributo é removido. Para escolher esse atributo a ser removido, são desenvolvidos modelos para vários subconjuntos criados, cada um desconsiderando um atributo. O conjunto de atributos do modelo com melhor desempenho é selecionado para a próxima iteração. Como esse método retira um atributo por vez, a seleção dos atributos considera a relação entre eles, sendo uma vantagem dessa abordagem [Chandrashekar and Sahin 2014].

3. Banco de dados

A base de dados escolhida para avaliação de carros neste estudo foi o Car Evaluation que se encontra disponível no repositório da Universidade da Califórnia Irvine em `https://archive-beta.ics.uci.edu/dataset/19/car+evaluation` já em formato *one-hot encoding*. Na Tabela 1, é apresentado um detalhamento dos atributos presentes no banco de dados. Como o banco de dados foi disponibilizado em formato *one-hot encoding*, cada um dos valores possíveis para o atributo representa um novo atributo com valor binário.

Tabela 1. Apresentação dos atributos presentes no banco de dados.

Atributo	Valores possíveis
Preço de Compra	Baixo, Médio, Alto e Muito Alto
Custo da Manutenção	Baixo, Médio, Alto e Muito Alto
Número de portas	Duas, Três, Quatro e Cinco ou mais
Número de passageiros	Dois, Quatro, Cinco ou mais
Tamanho do porta malas	Pequeno, Médio e Grande
Segurança	Baixa, Média e Alta

O banco de dados possui 1728 instâncias no total, sendo 21 atributos com valores binários que, para um melhor entendimento, podem ser agrupados em 6 categorias, e há 4 classes distintas: inaceitável, aceitável, bom e muito bom. Dessas instâncias, 1210 pertencem à classe "inaceitável", 384 pertencem à classe "aceitável", 65 pertencem à classe "bom" e 69 pertencem à classe "muito bom". Além disso, na Figura 1, pode-se observar o desbalanceamento das classes, além da visualização da separação do *dataset* em *folds* estratificados para a realização da validação cruzada *k-fold* com k=5.

4. Experimentos

Neste trabalho, foi realizada a comparação dos modelos *K-Nearest Neigbor* (KNN), Regressão Logística (RL), *Naive Bayes* (NB), Árvore de Decisão (DT), Perceptron Multi Camadas (PMC), *Support Vector Machine* (SVM), Floresta Aleatória (RF) e *Gradient Boosting* (GB). Neste experimento foi utilizado seleção de hiperparâmetros, balanceamento de instâncias e seleção de atributos, seguindo os passos apresentados no pseudocódigo da Tabela 2. Para os métodos de balanceamento foi utilizado a biblioteca Imbalanced Learning [Lemaître et al. 2017] e para a implementação dos modelos, o ajuste de hiperparâmetros e a seleção de atributos foi utilizado a biblioteca Scikit-Learn [Pedregosa et al. 2011].

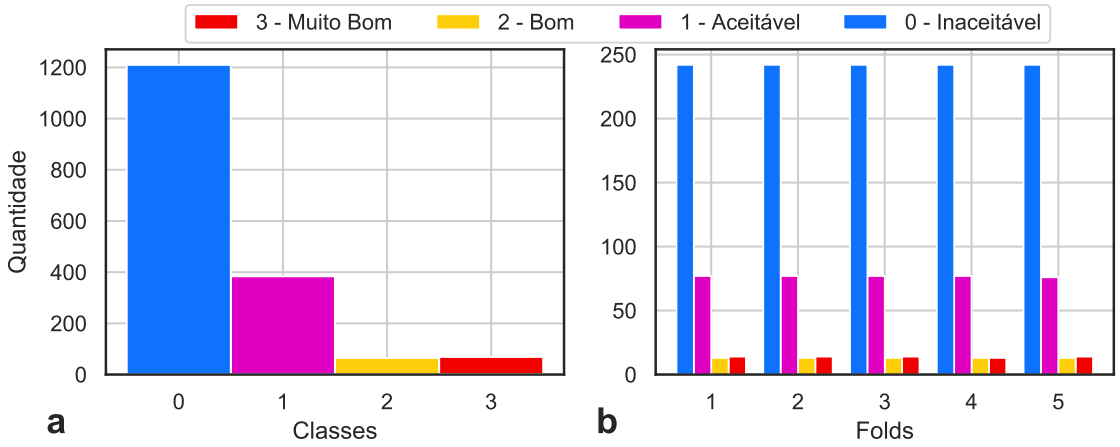

Figura 1. Apresentação da distribuição das classes do *dataset*. a) Divisão originalmente presente no *dataset*. b) Divisão do *dataset* em *folds* estratificados.

Tabela 2. Pseudocódigo do algoritmo utilizado para o treino e teste dos modelos.

Ajuste dos modelos
for modelo **in** [‘KNN’, ‘RL’, ‘BNB’, ‘DT’, ‘PMC’, ‘SVM’, ‘RF’, ‘GB’]:
crie um dataframe para receber os registros de cada modelo
for fold **in** [1, 2, 3, 4, 5]:
separe os dados de treinamento e teste
for balanceamento **in** [‘Sem’,‘Subamostragem’, ‘Sobreamostragem’, ‘Híbrido’]:
faça o balanceamento dos dados de treinamento
selecione os melhores hiperparâmetros com busca exaustiva e validação 5-fold
for N_atributos **in** [‘Todos’,16, 11, 6]:
selecione os melhores N_atributos com validação 5-fold
treine o melhor modelo com os melhores atributos
teste o modelo com os melhores atributos
calcule F1-*Score*, Acurácia e ROC-AUC do teste
armazene no dataframe o modelo, o fold, os atributos e o desempenho no teste

4.1. Balanceamento

Neste estudo, foram realizados três tipos de balanceamento: subamostragem, sobreamostragem e híbrido. O balanceamento foi realizado apenas no conjunto de treino e o teste foi preservado sem balanceamento, para todas as técnicas avaliadas.

Após a aplicação de subamostragem aleatória no conjunto de treino, todas as classes, exceto a minoritária (classe 2), foram aleatoriamente reduzidos até atingirem o total de 52 instâncias cada. Para a sobreamostragem, após a aplicação do SMOTE no conjunto de treino, exceto a classe majoritária (classe 0), todas as classes foram populadas até obterem 968 instâncias cada. Para obter um balanceamento híbrido, foi realizada a redução parcial da classe 0 para 450 instâncias e o aumento das demais classes até atingirem o total de 450 instâncias cada.

As estratégias de balanceamento foram aplicadas previamente à seleção de hiper-

parâmetros e atributos (Figura 2a), ou seja, elas foram aplicadas em cada conjunto de treinamento da validação *k-fold* externa (Figura 2b).

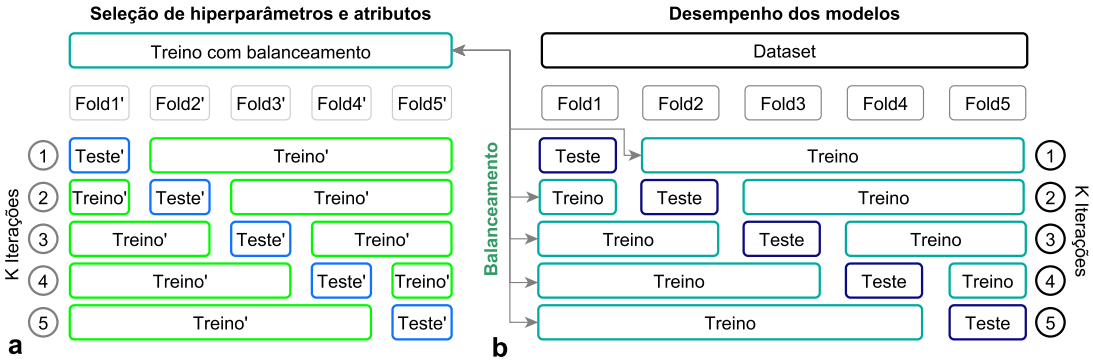

Figura 2. Validação cruzada 5-*fold* estratificada para avaliação das estrategias de balanceamento. a) Seleção de hiperparâmetros e de atributos que ocorre para cada iteração da validação cruzada externa (Figura 2b) de cada técnica de balanceamento e modelo. b) Validação 5-*fold* para avaliar o desempenho da melhor combinação de hiperparâmetros e atributos para cada modelo técnica de balanceamento.

4.2. Seleção de hiperparâmetros

A seleção dos hiperparâmetros ocorreu em duas etapas: revisão da literatura e busca exaustiva. A primeira etapa teve o objetivo de identificar o principais hiperparâmetros que são ajustados para cada modelo, quais os hiperparâmetros são geralmente escolhidos para os hiperparâmetros categóricos e qual a faixa de variação para os hiperparâmetros contínuos. Na busca exaustiva, todas as combinações dos hiperparâmetros previamente selecionados foram avaliadas usando validação cruzada *k-fold* com k=5 para cada estratégia de balanceamento, conforme apresentado na Figura 2b. Na Tabela 3, são apresentados os hiperparâmetros avaliados para cada modelo e os trabalhos que nortearam a escolha do espaço de busca. Para os hiperparâmetros não mencionados, foi mantido o padrão da biblioteca *scikit-learn*. Além disso, para o DT foram adotados os mesmos hiperparâmetros utilizados para RF, quando aplicável. Por fim, neste estudo, o modelo NB utilizado o *Bernoulli Naive Bayes*.

4.3. Seleção de atributos

A seleção de atributos foi realizada com o método SBS utilizando os hiperparâmetros ajustados para cada estratégia de balanceamento. Os modelos foram avaliados com quatro números possíveis de atributos: 21 (todos), 16, 11 e 5. A seleção dos atributos foi realizada com validação 5-*fold* estratificada, conforme apresentado na Figura 2a.

4.4. Avaliação do desempenho dos modelos

O desempenho dos modelos ajustados durante a validação 5-*fold* foi obtido usando média e desvio padrão da acurácia, F1-*Score*, área abaixo da curva ROC (AUC-ROC) e matriz de confusão. Além disso, a seleção da melhor combinação de hiperparâmetros de cada modelo por *fold* foi feita a partir da acurácia.

Tabela 3. Hiperparâmetros avaliados na busca exaustiva.

Modelo	Hiperparâmetros	Referência
BNB	alpha $\in \{0, 0.05, 0.1, ...0.95\}$	[Ramamohan et al. 2022]
DT	max_depth $\in \{5, 10, 15, ...45\}$ criterion $\in \{`gini', `entropy'\}$	[Yang and Shami 2020]
GB	n_estimators $\in \{10, 100\}$ max_depth $\in \{2, 3, 5, 7, 10\}$	[Bentéjac et al. 2021]
KNN	n_neighbors $\in \{5, 10, 15, ...45\}$ weights $\in \{2, 3, 5, 7, 10\}$ metric $\in \{`manhattan', `euclidean'\}$	[Bentéjac et al. 2021]
PMC	hidden_layer_size $\in \{5, 10, 15, ...45\}$ learning_rate_init $\in \{0.001, 0.01, 0.05\}$ learning_rate $\in \{`constant', `adaptive'\}$ activation $\in \{`relu', `tanh'\}$	[Ramamohan et al. 2022]
RF	n_estimators $\in \{10, 100\}$ max_depth $\in \{5., 10, 15, ...45\}$ criterion $\in \{`gini', `entropy'\}$	[Yang and Shami 2020]
RL	penalty $\in \{`l1', `l2', `elasticnet'\}$ l1_ratio $\in \{0, 1\}$ tol $\in \{0.00001, 0.01\}$ warm_start $\in \{True, False\}$	[Arafa et al. 2022]
SVM	C $\in \{0.1, 1, 10\}$ kernel $\in \{'linear', ' poly', ' rbf'\}$ tol $\in \{0.00001, 0.01\}$	[Yang and Shami 2020]

Todas as métricas utilizadas para avaliação dos modelos e hiperparâmetros são apresentadas a seguir. Para todas as métricas, VP representa os verdadeiros positivos; VN os verdadeiros negativos; FP os falsos positivos; e FN os falsos negativos.

Acurácia: A acurácia apresentada na Equação (1) descreve a fração das instâncias que foram classificadas corretamente. Em problemas multi-classe, essa métrica é extremamente sensível ao desbalanceamento de bancos de dados e deve ser ponderada pelo número de instâncias por classe ou acompanhada de outras métricas.

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN} \tag{1}$$

F1-*Score*: O F1-*Score* apresentado na Equação (4) é uma métrica recomendada para bancos de dados desbalanceados, uma vez que representa uma média harmônica entre a precisão apresentada na Equação (2) e o *recall* apresentado na Equação (3). Quando comparada com a acurácia, é mais informativa em relação ao desempenho dos modelos, no entanto, seu valor é mais difícil de ser interpretado.

$$Precisão = \frac{VP}{VP + FP} \tag{2}$$

$$Recall = \frac{VP}{VP + FN} \tag{3}$$

$$F1 - Score = 2 \times \frac{Precisão \times Recall}{Precisão + Recall} \tag{4}$$

ROC-AUC: A curva ROC (*Receiver Operating Characteristic Curve*) é uma forma gráfica de visualizar o desempenho de modelos de classificação binária a partir da relação entre a taxa de verdadeiros positivos e falsos positivos. Essa visualização também pode ser realizada para tarefas multi-classe, nesse caso, ela é mensurada comparando as classes par a par ou uma contra todas. A área abaixo da curva ROC (ROC-AUC) é uma alternativa para condensar a visualização em um único valor.

Matriz de confusão: A matriz de confusão é uma das principais formas de avaliação de modelos classificação, principalmente em tarefas multi-classe, uma vez que permite visualizar o número de amostras classificadas por classes e qual deveria ser a classe correta. Essa estratégia permite a visualização de classes com maior erro, identificar para quais classes o modelo possui maior dificuldade de discernir e permite identificar vieses na acurácia acarretada por bancos de dados desbalanceados.

5. Resultados

O repositório com os códigos utilizados para as análises apresentadas nesta seção será disponibilizado na versão final deste artigo para não revelar as identidades dos autores. As análises apresentadas nesta seção, bem como os códigos estão disponíveis em `https://github.com/lucas-fpaiva/AnalisePred_I/tree/main/Projeto/car_eval`

5.1. Comparação dos modelos sem técnicas de pré-processamento

Na Figura 3, são apresentados os desempenhos dos modelos avaliados para os *folds* de teste a partir de *boxplots*. Pode-se observar que todos os modelos tiveram acurácia próxima ou superior a 90%, o que poderia indicar que todos os modelos avaliados foram capazes de prever a aceitação dos carros. Essa suposição parece ser reforçada quando se olha para a ROC-AUC, com todos os modelos com área média superior a 96%. No entanto, o F1-*Score* apresenta um panorama diferente, pois os modelos NB, KNN e RL aparecem com desempenhos inferiores a 80%.

Apesar das divergências entre as métricas para alguns modelos, pode-se observar que os modelos DT, PMC, SVM e GB, possuem desempenho superior a 90% em todas as métricas, o que sugere que de fato estes modelos se ajustam bem ao problema de classificação de aceitação de carro. Especialmente, os modelos PMC e SVM, com desempenho superior a 99% e desvios padrões inferiores a 1% nas três métricas.

Na Figura 4, são apresentadas as matrizes de confusão para cada modelo. Em cada matriz, são apresentadas as médias dos desempenhos para os cinco *folds* de teste, onde nas diagonais principais são apresentadas as acurácias por classe. Pode-se observar que a classe 3 (muito bom) foi a que proporcionou maior dificuldade para os modelos,

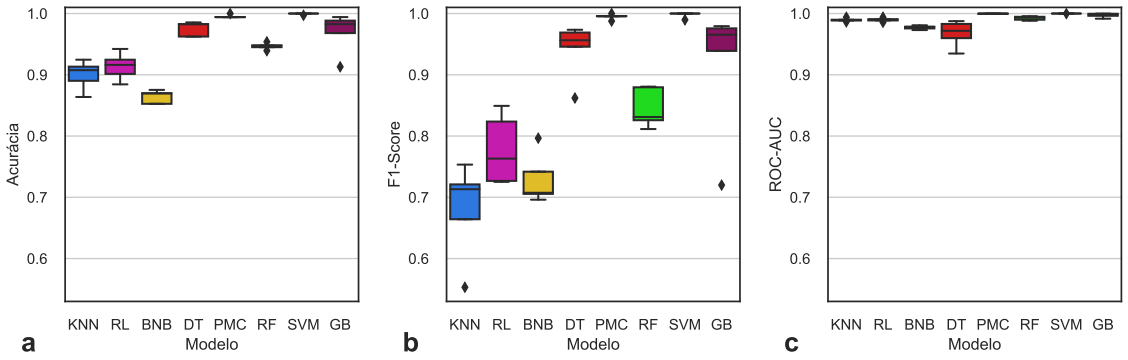

Figura 3. Comparação dos modelos avaliados em função dos desempenhos por *fold* sem balanceamento dos dados de treinamento e seleção de atributos.

apresentando acurácia menor que 15% para o KNN, acompanhado de RL e NB com acurácia inferior a 40%.

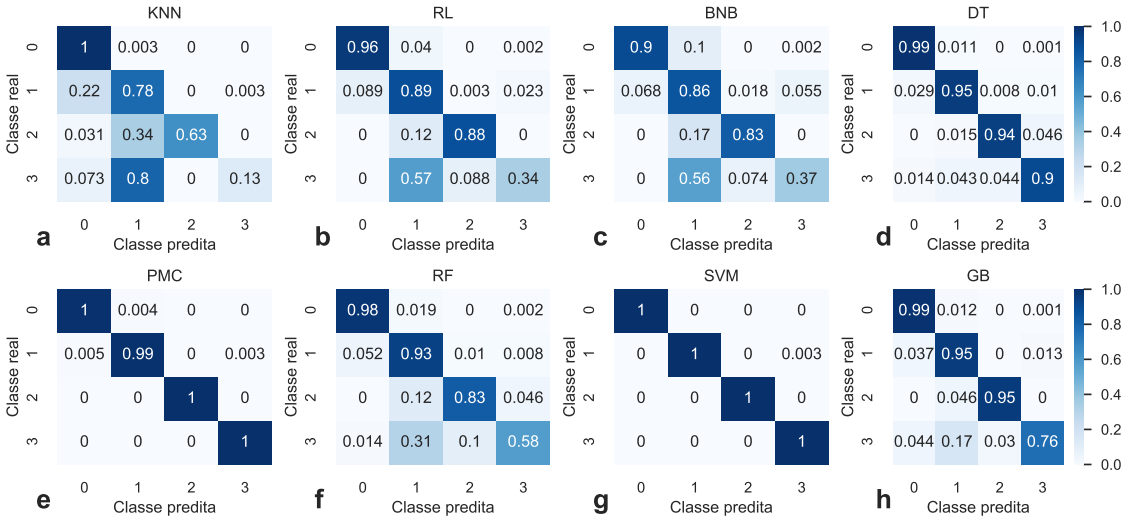

Figura 4. Matriz de confusão média para todos os algoritmos. a) *K-Nearest Neighbor*; b) Regressão Logística; c) *Naive Bayes*; d) Árvore de Decisão; e) Perceptron Multi Camadas; f) *Support Vector Machine*; g) Floresta Aleatória; h) *Gradient Boosting*.

Vale destacar que essa classe possui baixo número de instâncias, cerca de 16 por *fold*, enquanto que a classe 0 (inaceitável), possui 242 instâncias por *fold*. Esse reduzido número de instâncias na classe 3 justifica o baixo desempenho de alguns modelos. Além disso, o erro dessa classe é minimizado quando se olha para a acurácia global dos modelos, devido a sua baixa representatividade. Por fim, a matriz de confusão reafirma a superioridade dos modelos SVM e PMC para a classificação da aceitabilidade de carros em bancos de dados desbalanceados.

5.2. Efeito do uso de balanceamento e seleção de atributos

Na Figura 5, é apresentado o F1-*Score* médio de todos os modelos para todas as estratégias de pré-processamento. Essa métrica foi escolhida para a seleção dos melhores modelos, pois foi a que melhor retratou o desempenho dos modelos para esse banco de dados.

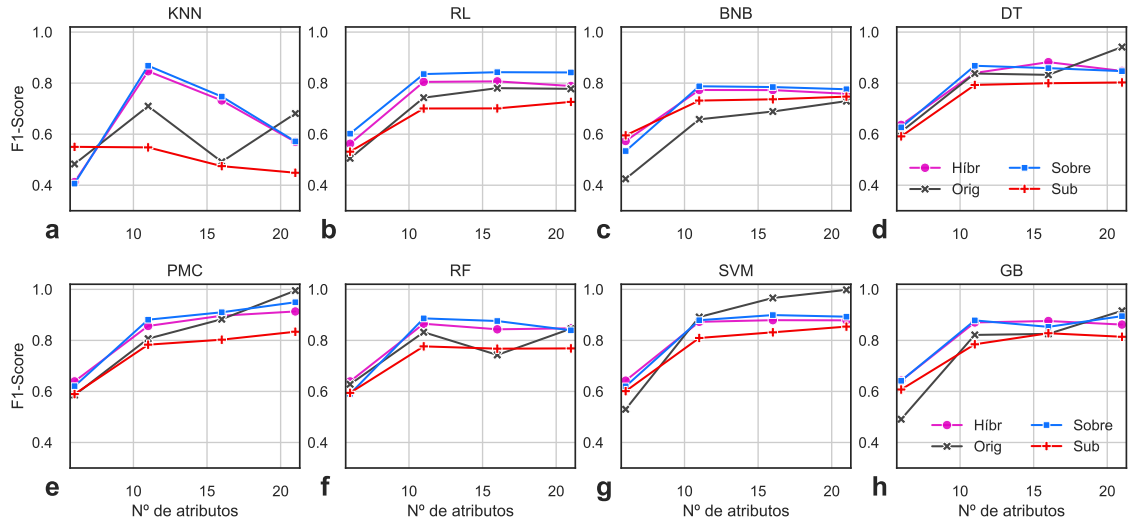

Figura 5. Comparação do desempenho no F1-*Score* dos modelos avaliados.

É possível observar que, em geral, os modelos desenvolvidos com sobreamostragem obtiveram melhores resultados, apresentando comportamento semelhante ao balanceamento híbrido. Enquanto que os modelos treinados com subamostragem apresentaram F1-*Score* inferior para a maioria dos algoritmos. Apesar da sobreamostragem mostrar melhor F1-*Score* para a maioria dos modelos, quando são utilizados todos os 21 atributos, ou seja, sem seleção de atributos, o desempenho só é melhor que não fazer o balanceamento para os algoritmos RL e BNB.

Em relação à seleção de atributos, para metade dos algoritmos (KNN, RL, BNB e RF), foi obtido o melhor F1-*Score* ao reduzir o número de atributos. No entanto, o modelo com maior desempenho dentre quatro algoritmos que proveram os melhores resultados gerais (SVM, PMC, DT e GB) foi obtido usando todos os 21 atributos e nenhum balanceamento. Além disso, em geral, há uma redução mais acentuada na performance quando o número de atributos é reduzido de 11 para 6.

Analisando a Tabela 4, pode-se observar que o uso combinado de sobreamostragem e seleção atributos aumenta o F1-*Score* dos algoritmos RF, RL, KNN e BNB. O maior aumento foi para o KNN, com aumento de 18 pontos percentuais (pp) no F1-*Score* e 5pp na acurácia. Apesar disso, os melhores algoritmos continuaram sendo SVM, PMC, DT e GB, desenvolvidos sem nenhuma das combinações de balanceamento e seleção de atributos. Para os modelos SVM e PMC, isso pode ter ocorrido por esses modelos já terem alcançado desempenhos superiores a 99% sem pré-processamento.

Na Figura 6, são apresentadas as matrizes de confusão da diferença entre as predições dos modelos com as melhores combinações de pré-processamento e os modelos sem balanceamento e seleção de atributos. As matrizes são apresentadas apenas para

Tabela 4. Média e desvio padrão de todos os modelos quando treinados com os melhores hiperparâmetros para os 5 *folds* usados para teste.

Modelo	Balanceamento	Atributos	F1-*Score* (s)	Acurácia (s)	ROC-AUC (s)
SVM	**No**	**21**	**0,998 (0,005)**	**0,999 (0,001)**	**1.000 (0.000)**
PMC	**No**	**21**	**0,995 (0,005)**	**0,995 (0,003)**	**1,000 (0,000)**
DT	No	21	0,941 (0,046)	0,975 (0,012)	0,967 (0,021)
GB	No	21	0,916 (0,111)	0,969 (0,033)	0,997 (0,003)
RF	No	21	0,846 (0,032)	0,947 (0,005)	0,992 (0,003)
RF	Over	11	**0,886 (0,034)**	0,951 (0,011)	0,987 (0,013)
RL	No	21	0,778 (0,057)	0,914 (0,022)	0,990 (0,003)
RL	Over	16	**0,843 (0,049)**	0,911 (0,021)	0,986 (0,004)
KNN	No	21	0,681 (0,078)	0,900 (0,024)	0,990 (0,003)
KNN	Over	11	**0,868 (0,060)**	**0,950 (0,015)**	0,977 (0,005)
BNB	No	21	0,729 (0,041)	0,864 (0,011)	0,977 (0,003)
BNB	Over	11	**0,788 (0,043)**	0,844 (0,016)	0,969 (0,006)

os algoritmos que melhoraram o desempenho com as abordagens de pré-processamento avaliadas, permitindo observar as melhorias por cada classe.

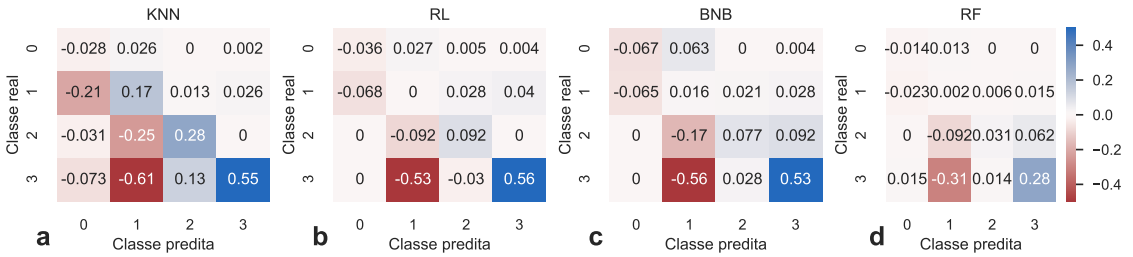

Figura 6. Matriz de confusão da diferença entre os modelos com e sem balanceamento e seleção de atributos para todos os algoritmos que melhoraram o desempenho com balanceamento e seleção de atributos. a) *K-Nearest Neighbor*; b) Regressão Logística; c) *Naive Bayes*; d) *Random Forest*; h) *Gradient Boosting*.

Nos casos que o balanceamento melhorou o resultado, ele melhorou a acuracidade principalmente da classe 3 (entre 28 e 56pp), que tinha obtido pior resultado por ter poucas instâncias, e reduziu o desempenho da classe 0 (entre 1 e 7pp). Esse efeito é esperado, pois o balanceamento reduz representatividade da classe majoritária. É importante destacar que a redução do erro da classe 3 se deve à diminuição de sua classificação errônea como classe 1 (diminuição entre 31 e 61pp). Um erro relevante, uma vez que a classe 3 (muito bom) está distante da classe 1 (aceitável).

O KNN teve uma pequena melhora na acurácia pois, antes do balanceamento, o modelo acertava com facilidade a classe majoritária, mas tinha baixo desempenho para as classes minoritárias. Devido a isso, com o balanceamento, houve uma pequena melhora na acurácia, entretanto uma melhora considerável do F1-*Score*, indicando assim uma melhora no acerto das predições de um modo geral. Para o KNN, as classes 1 e 2 também tiveram melhoria maior que 15pp.

Notem que, mesmo com o aumento do F1-*Score* para alguns algoritmos (Tabela 4), a acurácia diminui para o BNB e RL por causa da redução da performance de predição da classe 0 (como visto na Figura 6), que é majoritária, tendo maior peso no cálculo da acurácia.

5.3. Desempenho e hiperparâmetros dos melhores modelos por *fold*

Para cada iteração da validação cruzada, foi realizada uma busca exaustiva no espaço de busca previamente definido. Devido a isso, frequentemente foram encontradas combinações diferentes de hiperparâmetros de um modelo para cada *fold* de teste. Na Tabela 5, são apresentados os hiperparâmetros selecionados por *fold* e o desempenho dos modelos para cada *fold*. Pode-se observar que os modelos tiveram desempenho superior a 99% para todos os *folds*, apesar da variação dos dados de treino e teste e da varição dos hiperparâmetros.

Tabela 5. Hiperparâmetros selecionados por *fold* e desempenho no teste de cada modelo para a respectiva combinação de hiperparâmetros.

Modelo	Hiperparâmetros	Acurácia	F1-*Score*	ROC-AUC	*Fold*
SVM	kernel: 'rbf'	1,000	1,000	1,000	1
	kernel: 'rbf'	0,997	0,990	1,000	2
	kernel: 'poly'	1,000	1,000	1,000	3
	kernel: 'poly'	1,000	1,000	1,000	4
	kernel: 'rbf'	1,000	1,000	1,000	5
	C: 10				Todos
PMC	hls*:30, lri*: 0,05	1,000	1,000	1,000	1
	hls:15, lri: 0,05	0,994	0,988	1,000	2
	hls:25, lri: 0,05	0,994	0,996	0,999	3
	hls:10, lri: 0,01	0,994	0,996	1,000	4
	hls:10, lri: 0,01	0,994	0,996	1,000	5
	activation: 'relu', lr*:'constant'				Todos

*Nota: hls=hidden_layer_size, lri=learning_rate_init e lr=learning_rate.

Quando comparado com o PMC, o SVM foi ligeiramente mais estável em relação à variação dos hiperparâmetros e em relação ao desempenho. Em caso de uma aplicação usando o SVM em um cenário real, a forma mais fácil de selecionar os hiperparâmetros seria utilizar as combinações que mais se repetiram. Neste caso, o modelo poderia ser retreinado com todo o banco de dados utilizando C=10 e *kernel*='rbf'.

5.4. Comparação com a literatura

Na Tabela 6, são apresentados trabalhos que utilizaram modelos de aprendizado de máquina para o mesmo banco de dados de avaliação da aceitação de carros. Pode-se observar que os resultados encontrados neste estudo são coerentes com a literatura para todos os algoritmos comparados. Além disso, DT e SVM obtiveram acurácias superiores ao estado da arte, enquanto que o modelo proposto PMC obteve acurácia igual à maior acurácia encontrada por um modelo PMC em [Makki et al. 2011]. Por fim, vale destacar que o modelo SVM ajustado neste trabalho apresenta acurácia maior que todos os modelos de aprendizado de máquina encontrados na literatura.

Tabela 6. Comparação com os trabalhos da literatura para o mesmo banco de dados de avaliação de carros. Vale destacar que resultados não são diretamente comparáveis, pois podem ter sido obtidos com divisões diferentes dos dados.

Modelo	Acurácia	Referência
NB	**0,935**	[Awwalu et al. 2014]
NB	0,823	[Rehman et al. 2018]
NB	0,858	[Makki et al. 2011]
NB	0,864	[Jain and Vishwakarma 2017]
NB	0,844	Este trabalho
DT	0,932	[Awwalu et al. 2014]
DT	0,911	[Jain and Vishwakarma 2017]
DT	**0,975**	Este trabalho
GB	**0,994**	[Awwalu et al. 2014]
GB	0,969	Este trabalho
KNN	**0,954**	[Uzut and Buyrukoğlu 2020]
KNN	0,779	[Jain and Vishwakarma 2017]
KNN	0,950	Este trabalho
PMC	0,935	[Awwalu et al. 2014]
PMC	0,930	[Uzut and Buyrukoğlu 2020]
PMC	0,949	[Rehman et al. 2018]
PMC	**0,995**	[Makki et al. 2011]
PMC	0,810	[Potdar et al. 2017]
PMC	**0,995**	Este trabalho
RF	**0,979**	[Uzut and Buyrukoğlu 2020]
RF	0,812	[Jain and Vishwakarma 2017]
RF	0,951	Este trabalho
SVM	0,989	[Uzut and Buyrukoğlu 2020]
SVM	**0,999**	Este trabalho

6. Conclusões

Neste trabalho, foi utilizado aprendizado de máquinas para a classificação da aceitabilidade de carros em um *dataset* desbalanceado baseado em características de conforto, espaço, segurança, custos de manutenção e valor de compra. Foram avaliadas técnicas de balanceamento de instância para oito modelos comumente utilizados na literatura. Os modelos foram ajustados utilizando busca exaustiva de hiperparâmetros e seleção de atributos.

A combinação de seleção de atributos e *oversampling* fez com que metade dos algoritmos aumentassem seu desempenho, o destaque foi o modelo KNN, com aumento de 18pp do F1-*Score* e 5pp de acurácia. Mesmo com o aumento do desempenho, os melhores algoritmos foram o SVM e o PMC, alcançando mais de 99% de acurácia e F1-*Score*, sem balanceamento e seleção de atributos. Quando comparada com a literatura, a acurácia do SVM foi superior ao estado da arte. Além disso, foi observada a necessidade de utilização de várias métricas, especialmente as que avaliam o desempenho por classe, para avaliações mais completas dos modelos.

A maior limitação deste trabalho é a utilização de somente um banco de dados para avaliação dos modelos. Em trabalhos futuros, outros bancos de dados desbalanceados podem ser utilizados visando levantar quais métricas, modelos e estratégias de processamento são mais eficientes para esse tipo de problema.

Referências

[Arafa et al. 2022] Arafa, A., Radad, M., El-Fishawy, N., and Badawy, M. (2022). Logistic regression hyperparameter optimization for cancer classification. *Menoufia Journal of Electronic Engineering Research*, 31(1):1–8.

[Awwalu et al. 2014] Awwalu, J., Ghazvini, A., and Bakar, A. A. (2014). Performance comparison of data mining algorithms: a case study on car evaluation dataset. *International Journal of Computer Trends and Technology (IJCTT)*, 13(2):78–82.

[Bentéjac et al. 2021] Bentéjac, C., Csörgő, A., and Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, 54(3):1937–1967.

[Chandrashekar and Sahin 2014] Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28.

[Chawla et al. 2002] 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.

[De Diego et al. 2022] De Diego, I. M., Redondo, A. R., Fernández, R. R., Navarro, J., and Moguerza, J. M. (2022). General performance score for classification problems. *Applied Intelligence*, 52(10):12049–12063.

[Ferreira-Paiva et al. 2022] Ferreira-Paiva, L., Lopes, H. G., Alfaro-Espinoza, E. R., Félix, L. B., and Neves, R. V. A. (2022). Towards a device for helping deaf people to dance: estimation of forro bar length using artificial neural network. *IEEE Latin America Transactions*, 20(6):970–976.

[Guyon and Elisseeff 2003] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.

[Jain and Vishwakarma 2017] Jain, P. and Vishwakarma, S. K. (2017). A case study on car evaluation and prediction: comparative analysis using data mining models. *International Journal of Computer Applications (0975–8887)*, 172(9).

[Jović et al. 2015] Jović, A., Brkić, K., and Bogunović, N. (2015). A review of feature selection methods with applications. In *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 1200–1205, Opatija. IEEE.

[Krawczyk 2016] Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232.

[Lemaître et al. 2017] Lemaître, G., Nogueira, F., and Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5.

[Lopez-Vazquez et al. 2020] Lopez-Vazquez, V., Lopez-Guede, J. M., Marini, S., Fanelli, E., Johnsen, E., and Aguzzi, J. (2020). Video image enhancement and machine learning pipeline for underwater animal detection and classification at cabled observatories. *Sensors*, 20(3):726.

[Maier et al. 2019] Maier, A., Syben, C., Lasser, T., and Riess, C. (2019). A gentle introduction to deep learning in medical image processing. *Zeitschrift für Medizinische Physik*, 29(2):86–101.

[Makki et al. 2011] Makki, S., Mustapha, A., Kassim, J., Gharayebeh, E., and Alhazmi, M. (2011). Employing neural network and naive bayesian classifier in mining data for car evaluation. In *Proc. ICGST AIML-11 Conference*, pages 113–119.

[Pedregosa et al. 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Potdar et al. 2017] Potdar, K., Pardawala, T. S., and Pai, C. D. (2017). A comparative study of categorical variable encoding techniques for neural network classifiers. *International Journal of Computer Applications*, 175:7–9.

[Ramamohan et al. 2022] Ramamohan, V., Singhal, S., Gupta, A. R., and Bolia, N. B. (2022). Discrete simulation optimization for tuning machine learning method hyperparameters. *arXiv preprint arXiv:2201.05978*.

[Rehman et al. 2018] Rehman, Z. U., Fayyaz, H., Shah, A. A., Aslam, N., Hanif, M., and Abbas, S. (2018). Performance evaluation of mlpnn and nb: a comparative study on car evaluation dataset. *International Journal of Computer Science and Network Security*, 18(9):144–147.

[Shruthi et al. 2019] Shruthi, U., Nagaveni, V., and Raghavendra, B. (2019). A review on machine learning classification techniques for plant disease detection. In *2019 5th International conference on advanced computing & communication systems (ICACCS)*, pages 281–284. IEEE.

[Uzut and Buyrukoğlu 2020] Uzut, Ö. G. and Buyrukoğlu, S. (2020). Hyperparameter optimization of data mining algorithms on car evaluation dataset. *Euroasia Journal of Mathematics, Engineering, Natural & Medical Sciences*, 8(9):70–76.

[Yang and Shami 2020] Yang, L. and Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415:295–316.

[Yu et al. 2022] Yu, B., Li, C., Mirza, N., and Umar, M. (2022). Forecasting credit ratings of decarbonized firms: Comparative assessment of machine learning models. *Technological Forecasting and Social Change*, 174:121255.