Melhorando o desempenho da árvore de decisão utilizando *Bagging Ensemble* na identificação de doenças cardíacas em conjuntos de dados desbalanceados

Laura Damaceno de Almeida[1]

[1]Centro de Matemática, Computação e Cognição, Universidade Federal do ABC Av.dos Estados, 5001 - Bairro S. Terezinha, Santo André, SP, Brasil - CEP: 09210-580

laura.da.almeida@hotmail.com

Abstract. *Cardiovascular problems are the number 1 cause of death in the world and recent studies have used supervised machine learning with imbalanced data to predict the chances of heart disease and rank the patient's risk level. However, imbalanced data is a common problem in applying this type of machine learning. In this article, the Bagging Ensemble technique will be used to improve the performance of the decision tree model in the study of [Rajdhan et al. 2020].*

Resumo. *Problemas cardiovasculares são o número 1 na causa de mortes no mundo e estudos recentes utilizaram aprendizado de máquina supervisionado com dados desbalanceados para prever as chances de doença cardíaca e classificar o nível de risco do paciente. Entretanto classes desbalanceadas são problemas comuns na aplicação desse tipo de aprendizado de máquina. Portanto neste artigo será utilizado a técnica Bagging Ensemble para melhorar o desempenho do modelo de árvore de decisão obtido estudo de [Rajdhan et al. 2020].*

1. Introdução

A Organização Mundial da Saúde informou que problemas cardiovasculares (CVDs - *Cardiovascular Diseases*) é a causa número 1 de mortes no mundo, dentre elas ataques ou paradas cardíacas, e mais de 17.9 milhões de pessoas morrem devido a esse problema por ano.

Estudos recentes utilizaram técnicas de aprendizado de máquina como forma de predição de doenças cardíacas, e isso poderá auxiliar os profissionais de saúde a tomarem decisões mais precisas para a saúde dos pacientes. Entretanto esses estudos utilizaram conjuntos de dados com classes desbalanceadas, e isso é um problema ao utilizar aprendizado de máquina supervisionado, onde as classes não são representadas justamente [Arafat et al. 2019]. Com isto, o desempenho de um classificador tende a ser tendencioso em relação a algumas classes (classe majoritária) no conjunto de dados desequilibrado [Kaur et al. 2019]. As técnicas mais utilizadas para solucioná-lo são *under-sampling*, método que reduz randomicamente a classe marjoritária, *over-sampling* que é um método utilizado para gerar artificialmente novos dados da classe minoritária e os métodos *ensembles*. De acordo com [Dong et al. 2020] os métodos *ensemble classification* têm vantagens em termos de precisão, estabilidade e generalização, eles são amplamente adotados para resolver todos os tipos de problemas, como aprendizado de várias instâncias, aprendizado de múltiplos rótulos e aprendizado em classes desbalanceadas.

O estudo apresentado neste artigo irá trazer uma solução computacional para a melhoria do desempenho do algoritmo de árvore de decisão com base no estudo feito por [Rajdhan et al. 2020], na identificação de pacientes com doenças cardíacas.

Este artigo está organizado da seguinte sequência: o capítulo 2 aborda os trabalhos já existentes utilizados como base de fundamentação teórica do trabalho. No capítulo 3 é apresentado a fundamentação teórica. O capítulo 4 trás a proposta que será abordada neste artigo, juntamente com sua justificativa. O capítulo 5 aborda a metodologia utilizada nos experimentos do artigo. No capítulo 6 são apresentados os resultados dos experimentos realizados nos diferentes cenários propostos. Finalmente, no capítulo 7 são feitas as considerações finais, em que são apresentadas as conclusões relativas a este trabalho e são descritas algumas sugestões de trabalhos futuros.

2. Trabalhos relacionados

Nos estudos realizados com a aplicação de aprendizado de máquina para a identificação de problemas cardíacos, o algoritmo de árvore de decisão apresenta desempenho inferior comparado por exemplo com os algoritmo *Random Forest* e *Naive Bayes*.

Dentre os trabalhos realizados pode ser citado [Sujatha and Mahalakshmi 2020], que em seu estudo comparativo de algoritmos de classificação, o modelo de árvore de decisão obteve um dos piores valores de métricas: 78.9% de acurácia, 82.9% de precisão, 80.4% de *F1-Score*, 78.9% de *Area under the Curve* (AUC), nesse estudo foi implementado a técnica de Boruta para seleção de *features*. No estudo de [Rajdhan et al. 2020] o modelo de árvore de decisão foi o que obteve pior valor de acurácia, 81.9% de acurácia (Tabela 1.).

Tabela 1. Desempenho dos algoritmos utilizados na pesquisa de [Rajdhan et al. 2020]

Modelo	**Precisão**	***Recall***	***F1-Score***	**Acurácia**
Árvore de decisão	**84.5**	**82.3**	**83.5**	**81.9**
Random Forest	93.7	88.2	90.9	90.1
Regressão Logística	85.7	88.2	86.9	85.2
Naive Bayes	83.7	91.1	87.3	85.2

3. Fundamentação Teórica

Nos trabalhos apresentados anteriormente, foram utilizados aprendizado de máquina para predição de problemas cardíacos nos pacientes.

Uma árvore de decisão é um grafo acíclico direcionado em que cada nó ou é um nó de divisão, com dois ou mais sucessores, ou um nó folha [Lorena et al. 2000]. De acordo com [Rajdhan et al. 2020] o algoritmo é rápido, confiável, fácil de interpretar e com poucos preparação de dados é necessária.

De acordo com [Arafat et al. 2019] o método *ensemble* é o processo de combinar vários algoritmos classificadores para formar um algoritmo classificador forte para classificar novas instâncias com alta precisão de previsão. Os exemplos são *Random Forest*, *Bagging* e *Boosting*. De acordo com [Dong et al. 2020] modelos de classificação baseados em *ensemble* consiste 2 etapas:

1. Gerar resultados de classificação usando vários classificadores fracos, e
2. Integração de vários resultados em uma função de consistência para obtenha o resultado final com esquemas de votação.

O método *Bagging* gera subconjuntos de amostra por amostragem aleatória do conjunto de dados de treinamento e, em seguida, usa estes subconjuntos obtidos para treinar os modelos básicos para integração [Dong et al. 2020]. Neste presente trabalho será utilizado o algoritmo *BaggingClassifier* através da biblioteca *Scikit-learn*, biblioteca Python *open-source* mais proeminente para aprendizado de máquina [Müller and Guido 2016].

Além disso, em busca de um desempenho ótimo será realizado neste trabalho testes no algoritmo com diferentes valores nos parâmetros *max_samples* e *n_estimators*, pois de acordo com a documentação oficial da biblioteca *Scikit-learn* esses são os principais parâmetros a serem ajustados ao usar esse método. O parâmetro *n_estimators* é referente ao número de árvores na floresta, quanto maior o número melhor o desempenho, mas também levará a um custo computacional maior para calcular. O parâmetro *max_samples* é referente ao tamanho dos subconjuntos aleatórios de recursos a serem considerados ao dividir um nó. E de acordo com a biblioteca quanto menor o valor, maior será a redução da variância, mas também maior será o aumento do viés.

De acordo com [Lei 2020] o problema de seleção do parâmetro de ajuste está preocupado em encontrar o valor ideal do parâmetro, a partir de um determinado conjunto candidato finito, que leva ao menor risco preditivo. Nesse contexto, o *cross validation* encontra essencialmente o valor do parâmetro de ajuste cujos modelos ajustados têm pequeno risco preditivo. Portanto neste artigo, será utilizado esse método para fazer a seleção dos melhores valores dos parâmetros.

4. Proposta

Nos estudos apresentados na seção 2, utilizaram dados do repositório *UCI Machine Learning*, entretanto nas técnicas utilizadas pelos mesmos não consideraram realizar métodos *ensembles* para melhorar o desempenho dos algoritmos, em especial o de árvore de decisão que apresentou as piores métricas, por se tratarem de análises prévias do desempenhos dos algoritmos.

Um ponto interessante a ser destacado é que em ambos os estudos citados, o algoritmo *Random Forest* foi o que apresentou melhores resultados para as métricas avaliadas. De acordo com [Athey et al. 2019] esse algoritmo é classificado como um aprendizado de máquina supervisionado *ensemble*, que tem como algoritmo base para a construção do classificador a árvore de decisão. Portanto pode-se levantar a hipótese que aplicando o método *ensemble* pode-se ter uma melhora no valor da acurácia do modelo de árvore de decisão. Com essa hipótese em mente, a proposta apresentada neste *paper* será a implementação de *Bagging Ensemble* para melhorar o desempenho da árvore de decisão.

Conforme informado na seção 3, algoritmo utilizado foi o *BaggingClassifier*, que tem como algoritmo *default* a árvore de decisão. Foram testados diversos valores X no parâmetro *n_estimators*: 10, 50, 100, 500,1000 e 5000. À partir da implementação do *cross validation* com k=5, para cada um dos valores testados foram analisados os valores médios da acurácia e desvio padrão, e selecionado o valor para *n_estimators* que apresentou o melhor valor médio de acurácia e um valor baixo de desvio padrão [Harrison 2020].

Neste artigo será considerado X como o melhor valor encontrado para o parâmetro de *n_estimators* e Y como o melhor valor para o parâmetro de *max_samples*. Após a escolha do valor X, foram testados valores de 0.1 até 1.0 para o parâmetro *max_samples*, nesta etapa também foi utilizado o *cross validation* e o mesmo critério de avaliação foi aplicado (Figure 1).

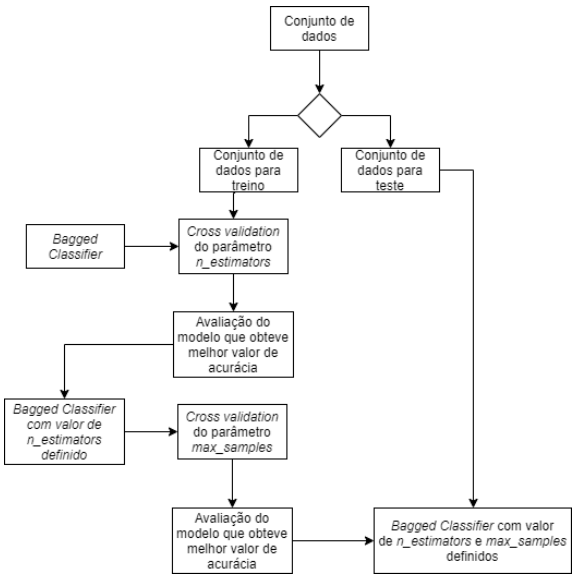

Figura 1. Processo de treinamento dos modelos

5. Metodologia

Foi utilizado neste presente trabalho o conjunto de dados *UCI Cleveland* do repositório *UCI Machine learning* que contém 14 atributos: *age, sex, CP, trestbps, chol, FBS, restecg, thalach, exang, oldpeak, slope, ca, thal, target* [Rajdhan et al. 2020].

O conjunto de dados foi divido em 80% para treino e 20% para teste, e os mesmos não passaram por nenhum tratamento, seguindo o mesmo processo adquiro nos experimentos de [Rajdhan et al. 2020].

Com os valores de X e Y definidos para os parâmetros, conforme apresentado na seção 4, o modelo final foi avaliado usando o conjunto de teste através das seguintes métricas de desempenho utilizada por [Rajdhan et al. 2020]: acurácia (porcentagem de classificações corretas), precisão (considerada uma medida da exatidão do classificador, uma vez que quantifica verdadeiras instâncias positivas entre todas as exemplos positivos [Picek et al. 2019]), *recall* (considerada uma medida do classificador completude, pois quantifica verdadeiras instâncias positivas que são encontradas entre instâncias positivas [Picek et al. 2019]), *F1 score* (média harmônica entre o *recall* e a precisão [Picek et al. 2019]).

6. Resultados e Análises

O conjunto de dados de treino, correspondente a 80% do conjunto original, é usado como entrada para realizar os experimentos nos valores dos parâmetros, na ordem mencionada na sessão 4 e exibido na Figura 1. E no primeiro teste para o parâmetros *n_estimators*

(Tabela 2. e Tabela 3.) foi identificado que o valor 1000 apresentou os melhores resultados para todas as métricas dentre os outros valores testados, portanto iremos considerá-lo como valor para o parâmetro.

Tabela 2. Desempenho da acurácia e precisão do algoritmo utilizando diferentes valores em *n_estimators*

N_estimators	Acuracia_media	Acuracia_std	Precisao_media	Precisao_std
10	0.785204	0.042490	0.815058	0.054046
50	0.797585	0.034256	0.800483	0.038266
100	0.795901	0.050340	0.801820	0.046867
500	0.804235	0.038872	0.804920	0.043907
1000	**0.807466**	**0.035821**	**0.810631**	**0.039756**
5000	0.800850	0.040917	0.803282	0.041992

Tabela 3. Desempenho do *recall e F1* algoritmo utilizando diferentes valores em *n_estimators*

N_estimators	*Recall_media*	*Recall_std*	F1_media	F1_std
10	0.778708	0.074243	0.793492	0.043303
50	0.829477	0.060572	0.813053	0.035391
100	0.821662	0.077263	0.809799	0.052019
500	0.838954	0.060892	0.819933	0.038093
1000	**0.837354**	**0.065184**	**0.821849**	**0.036282**
5000	0.832615	0.065053	0.816036	0.040331

Considerado o valor 1000 o parâmetro *n_estimators*, foi realizado um teste com diferentes valores para o parâmetro *max_samples* (Tabela 4. e Tabela 5.) e os valores 0.2 e 0.4 foram considerados os melhores dentre os outros testados, pois o valor médio da acurácia foi o melhor obtido e o desvio padrão foi um dos menores, portanto alcançaram assim o critério de avaliação considerado para este estudo.

Tabela 4. Desempenho da acurácia e precisão do algoritmo utilizando diferentes valores em *max_samples*, considerando *n_estimators*=1000

Max_samples	Acuracia_media	Acuracia_std	Precisao_media	Precisao_std
0.1	0.818299	0.057940	0.800273	0.056142
0.2	**0.822330**	**0.050571**	**0.806516**	**0.051462**
0.3	0.819881	0.049333	0.806435	0.044283
0.4	**0.823163**	**0.047076**	**0.811961**	**0.041417**
0.5	0.817415	0.045560	0.807337	0.040731
0.6	0.815748	0.044713	0.809953	0.043261
0.7	0.814932	0.043289	0.808760	0.042017
0.8	0.811633	0.040001	0.811340	0.039269
0.9	0.806650	0.042081	0.805909	0.040884
1.0	0.799184	0.038141	0.802734	0.040876

Tabela 5. Desempenho do *recall* e F1 do algoritmo utilizando diferentes valores em *max_samples*, considerando *n_estimators*=1000

Max_samples	*Recall_media*	*Recall_std*	F1_media	F1_std
0.1	0.882277	0.061278	0.838206	0.050705
0.2	**0.880677**	**0.061602**	**0.840707**	**0.045496**
0.3	0.872923	0.067018	0.837236	0.046537
0.4	**0.871446**	**0.072824**	**0.839155**	**0.046039**
0.5	0.865292	0.067097	0.834061	0.043886
0.6	0.857477	0.067695	0.831501	0.043987
0.7	0.857415	0.069607	0.830731	0.043103
0.8	0.845108	0.062342	0.826460	0.039070
0.9	0.841908	0.068688	0.821872	0.042798
1.0	0.829415	0.064535	0.814156	0.038364

Com o dois valores dos parâmetros definidos para o modelo final (*n_estimators*=1000 e *max_samples*=0.2 e 0.4), eles foram finalmente testados com o conjunto de dados de teste, correspondente a 20% do conjunto original, ao final foi obtido 83.6% de acurácia, 82.5% de precisão, 91.6% de *recall* e 86.8% de *F1-Score* considerando 0.2 como valor pro *max_samples*. E alcançou 81.9% de acurácia, 82% de precisão, 88.8% de *recall* e 85.3% de *F1-Score* considerando 0.4 como valor pro *max_samples*. Portanto conclui-se que o melhor parâmetro é *n_estimators*=1000 e *max_samples*=0.2.

Além disto, pode-se perceber um ganho no valor de acurácia ao utilizar o método *Bagging*, pois no estudo apresentado por [Rajdhan et al. 2020] o modelo de árvore de decisão obteve 81.97%.

7. Conclusão

Problemas cardíacos têm se mostrado o maior causador de mortes do mundo, e estudos recentes aplicaram aprendizado de máquina como forma de predição e assim ajudar a tomada de decisão dos profissionais de saúde. Entretanto muitos conjuntos de dados utilizados para o treino desses modelos há um desbalanceamento entre classes, onde há mais amostras de pessoas que não têm problemas cardíacos do que pessoas que têm esse problema, e isso pode tornar os classificadores tendenciosos. Portanto foi considerado utilizar umas das estratégias para lidar com esse desbalanceamento, o *bagging ensemble* [Dong et al. 2020].

Modelos baseados em árvore são muito utilizados em problemas de classificação, e nos estudos realizados para detecção de problemas cardíacos, dentre eles [Rajdhan et al. 2020], foi o que obteve o pior desempenho alcançando 84.5% de precisão, 82.3% de *recall*, 83.5% de *F1-Score* e 81.9% de acurácia. Neste estudo foi abordado uma solução computacional para melhorar a acurácia desse algoritmo, utilizando o *bagging ensembles* com árvores de decisão.

Com a utilização do mesmo chegou-se em uma melhora não apenas na acurácia mas também no *recall* e *F1-Score*, com respectivamente os seguintes valores 83.6% de acurácia, 91.6% de *recall* e 86.8% de *F1-Score*. Portanto com esse ajuste percebe-se que

ajudou o modelo a diferenciar as classes e conseguir com maior exatidão a identificação de pessoas com problemas cardíacos.

Como houve uma melhora para o algoritmo de árvore de decisão juntamente com o *bagging ensemble*, pode-se considerar como trabalho futuro utilizar outros algoritmos, como: KNN, *naive bayes* e regressão logística, e analisar para ver se os mesmos obterão bons resultados.

Referências

Arafat, M. Y., Hoque, S., Xu, S., and Farid, D. M. (2019). Machine learning for mining imbalanced data. *IAENG International Journal of Computer Science*, 46(2):332–348.

Athey, S., Tibshirani, J., and Wager, S. (2019). Generalized random forests. *The Annals of Statistics*, 47(2):1148–1178.

Dong, X., Yu, Z., Cao, W., Shi, Y., and Ma, Q. (2020). A survey on ensemble learning. *Frontiers of Computer Science*, 14(2):241–258.

Harrison, M. (2020). *Machine Learning: Guia de Referência Rápida*. Novatec.

Kaur, H., Pannu, H. S., and Malhi, A. K. (2019). A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Computing Surveys (CSUR)*, 52(4):1–36.

Lei, J. (2020). Cross-validation with confidence. *Journal of the American Statistical Association*, 115(532):1978–1997.

Lorena, A. C., Gama, J., and Faceli, K. (2000). *Inteligência Artificial: Uma abordagem de aprendizado de máquina*. Grupo Gen-LTC.

Müller, A. C. and Guido, S. (2016). *Introduction to machine learning with Python: a guide for data scientists*. "O'Reilly Media, Inc.".

Picek, S., Heuser, A., Jovic, A., Bhasin, S., and Regazzoni, F. (2019). The curse of class imbalance and conflicting metrics with machine learning for side-channel evaluations. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019(1):1–29.

Rajdhan, A., Agarwal, A., Sai, M., Ravi, D., and Ghuli, P. (2020). Heart disease prediction using machine learning.

Sujatha, P. and Mahalakshmi, K. (2020). Performance evaluation of supervised machine learning algorithms in prediction of heart disease. In *2020 IEEE International Conference for Innovation in Technology (INOCON)*, pages 1–7. IEEE.