

Comparação experimental entre técnicas de aprendizado aplicadas a problemas de classificação

Pedro Anselmo Santana De Angeli*

Abstract

Este artigo tem o objetivo de realizar uma comparação experimental entre um conjunto pré-definido de técnicas de aprendizado para classificação automática, baseadas na ideia de combinados de classificadores, aplicadas a alguns problemas de classificação. As técnicas escolhidas são: *Bagging*, *AdaBoost*, *Random Forest* e *Heterogeneous Pooling*¹. As bases de dados a serem utilizadas são *digits*, *wine* e *breast cancer*.

Keywords: Classificadores, Ensembles, Problemas de Classificação

1. Introdução

Um problema de classificação pode ser definido pela busca de um modelo capaz de receber como entrada um caso a ser avaliado e indicar qual classe ele pertence. São problemas de aprendizagem supervisionada, no qual é utilizado um conjunto de dados onde as classes são conhecidas para construir o modelo. Esse modelo, uma vez construído, poderá ser aplicado a novos dados para prever suas respectivas classes. Um *ensemble*, ou combinado de classificadores, consiste em um conjunto de classificadores treinados individualmente cujas decisões são combinadas de alguma forma.

Este artigo tem o objetivo de realizar uma comparação experimental entre quatro combinados de classificadores: *Bagging*, *AdaBoost*, *Random Forest* e *Heterogeneous Pooling*. Todos os classificadores foram treinados nas bases de dados

*Aluno do curso de Engenharia de Computação da Universidade Federal do Espírito Santo.

¹Combinado de classificadores composto por *Nearest Neighbors*, *Gaussian Naive Bayes* e *Decision tree*

digits, *wine* e *breast cancer*, que são disponibilizadas pela biblioteca *scikit-learn*.

A métrica utilizada para comparação foi a acurácia dos resultados obtidos. Para
15 cada base, o procedimento experimental de treinamento foi realizado através de
3 rodadas de validação cruzada estratificada de ciclos aninhados de validação
e teste, com o ciclo interno de validação contendo 4 folds e o externo de teste
com 10 folds. Antes de cada treinamento, os dados são padronizados utilizando
a normalização com *z-score*.

20 2. Descrição dos métodos implementados

O único *ensemble* implementado foi o *Heterogeneous Pooling*. Os demais
foram obtidos pela biblioteca *scikit-learn*. Além disso, foi criada uma classe
com funções em comum para cada classificador, com o objetivo de modularizar
e facilitar a compreensão do código.

25 2.1. Métodos em comum

Construtor: O construtor de cada classe define todos os parâmetros necessários
para o treinamento. O modelo é instanciado, assim como o padronizador.
Também são definidos os hiperparâmetros que serão testados, o *grid search*
e a estratégia de validação cruzada.

30 ***fit()*:** Método utilizado para treinamento. É responsável pela chamada da
função do *scikit-learn* que fará o treinamento passando os parâmetros difinidos
no construtor.

***results()*:** Método que retorna a média, o desvio padrão e o intervalo de
confiança dos resultados obtidos no treinamento.

35 ***getScores()*:** Retorna as acurácias obtidas em cada fold de teste.

2.2. *Heterogeneous Pooling*

O *Heterogeneous Pooling* é um combinado de classificadores composto por
Nearest Neighbors, *Gaussian Naive Bayes* e *Decision tree*. Os três classificadores
foram obtidos pela biblioteca *scikit-learn*. O *ensemble* recebe um atributo
40 chamado *n_samples*, que representa a quantidade individual de cada um dos

três classificadores. Se $n_samples = 2$, por exemplo, o combinado será composto por 6 classificadores: 2 *Nearest Neighbors*, 2 *Gaussian Naive Bayes* e 2 *Decision tree*. Os métodos *fit()* e *predict()* descritos abaixo detalham o funcionamento do combinado.

45 ***fit()***: Primeiramente, são obtidas as frequências de cada classe na base de treino. Em seguida, cada um dos classificadores é treinado com uma base de treino de mesmo tamanho da original montada coletando aleatoriamente exemplos da base original com reposição.

50 ***predict()***: Cada exemplo é classificado por todos os classificadores. O resultado é selecionado utilizando votação majoritária. Se houver empate na votação, a classe mais frequente na base de treino é escolhida.

3. Descrição dos experimentos realizados

Antes de cada treinamento, os dados são padronizados utilizando a normalização com *z-score*. O procedimento experimental consiste em validação e teste 55 através de 3 rodadas de validação cruzada estratificada de ciclos aninhados, com o ciclo interno de validação contendo 4 folds e o externo de teste com 10 folds. A etapa de validação é feita pelo *grid search* considerando os seguintes hiperparâmetros:

Classificador	Hiperparâmetro
AdaBoost	n_estimators = [10, 25, 50, 100]
Bagging	n_estimators = [10, 25, 50, 100]
RandomForest	n_estimators = [10, 25, 50, 100]
HeterogeneousPooling	n_samples = [1, 3, 5, 7]

Table 1: Hiperparâmetros de cada classificador.

3.1. Digits

60 A base de dados *digits* possui 1797 amostras, 64 atributos e 10 classes. Cada classe possui aproximadamente 180 amostras. Os resultados obtidos após a execução do procedimento experimental são exibidos na Tabela 2.

Método	Média	Desvio Padrão	Limite Inferior	Limite Superior
AdaBoost	0.270	0.022	0.262	0.278
Bagging	0.954	0.014	0.949	0.959
RandomForest	0.976	0.009	0.973	0.979
HeterogeneousPooling	0.966	0.012	0.961	0.970

Table 2: Resultados obtidos na base Digits.

Da tabela, podemos perceber que o método *AdaBoost* obteve péssimo desempenho em relação aos demais, com uma acurácia muito baixa e um desvio
65 padrão maior que os outros.

O gráfico boxplot obtido para o experimento é exibido na Figura 1.

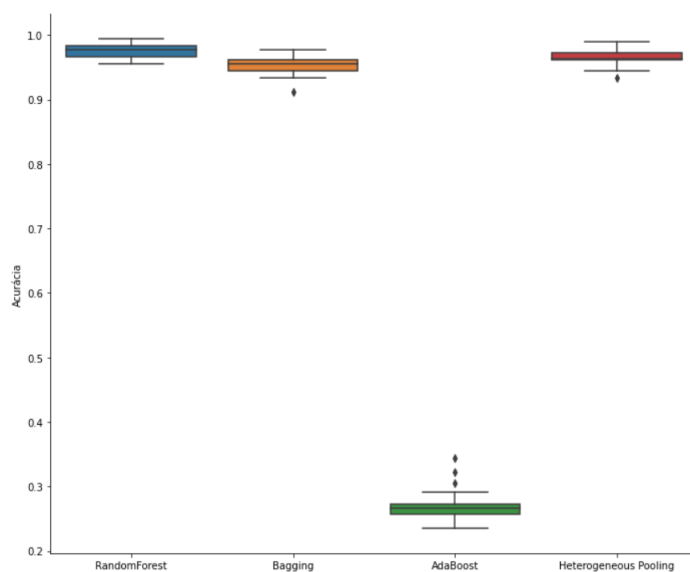


Figure 1: Boxplot obtido para a base digits.

Da tabela e do boxplot, nota-se que os métodos de melhor desempenho apresentaram comportamento muito semelhantes em termos de acurácia. Porém, essa avaliação não é suficiente para avaliar se um classificador é melhor do
70 que o outro. Para isso, foi utilizado testes estatísticos. A tabela 3 mostra na matriz triangular superior os resultados do teste t pareado e na matriz triangular

inferior os resultados do teste não paramétrico de wilcoxon.

Bagging	0.000	0.000	0.001
0.000	AdaBoost	0.000	0.000
0.000	0.000	RandomForest	0.000
0.001	0.000	0.000	Heterogeneous Pooling

Table 3: Testes de hipóteses na base digits.

As células em negrito representam os testes em que a hipótese nula foi rejeitada. Isso quer dizer que há diferença significativa entre os classificadores. Os demais aceitam a hipótese nula, logo não podemos concluir qual apresenta um melhor desempenho. Perceba que nesse caso todos os testes rejeitaram a hipótese nula.

3.2. Wine

A base de dados *wine* possui 178 amostras, 13 atributos e 3 classes. Cada classe possui 59, 71 e 48 amostras. Os resultados obtidos após a execução do procedimento experimental são exibidos na Tabela 4.

Método	Média	Desvio Padrão	Limite Inferior	Limite Superior
AdaBoost	0.914	0.076	0.887	0.941
Bagging	0.955	0.067	0.931	0.979
RandomForest	0.975	0.041	0.961	0.990
HeterogeneousPooling	0.966	0.061	0.944	0.988

Table 4: Resultados obtidos na base Wine.

Diferentemente da base *digits*, todos os classificadores obtiveram desempenho semelhante na base *wine*. O gráfico boxplot obtido para o experimento é exibido na Figura 2.

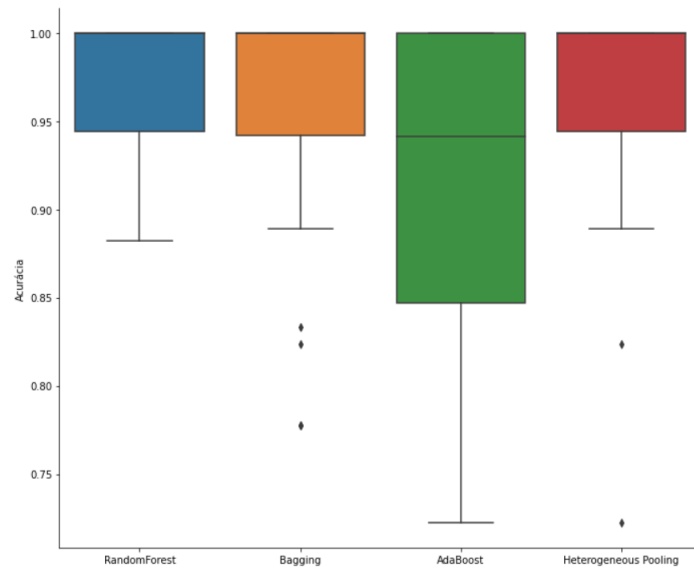


Figure 2: Boxplot obtido para a base wine.

85 A tabela 5 mostra na matriz triangular superior os resultados do teste t pareado e na matriz triangular inferior os resultados do teste não paramétrico de wilcoxon.

Bagging	0.047	0.055	0.287
0.043	AdaBoost	0.002	0.008
0.073	0.005	RandomForest	0.298
0.273	0.010	0.379	Heterogeneous Pooling

Table 5: Testes de hipóteses na base wine.

Diferentemente da base *digits*, nem todos os testes rejeitaram a hipótese nula.

90 3.3. Breast Cancer

A base de dados *breast cancer* possui 569 amostras, 30 atributos e 2 classes. Cada classe possui 212 e 357 amostras. Os resultados obtidos após a execução do procedimento experimental são exibidos na Tabela 6.

Método	Média	Desvio Padrão	Limite Inferior	Limite Superior
AdaBoost	0.968	0.024	0.959	0.976
Bagging	0.958	0.025	0.949	0.967
RandomForest	0.960	0.024	0.951	0.969
HeterogeneousPooling	0.960	0.024	0.951	0.968

Table 6: Resultados obitdos na base Breast Cancer.

Assim como na base *wine*, os modelos se comportaram de maneira semel-
95 hante na base *breast cancer*. O gráfico boxplot obtido para o experimento é exibido na Figura 3.

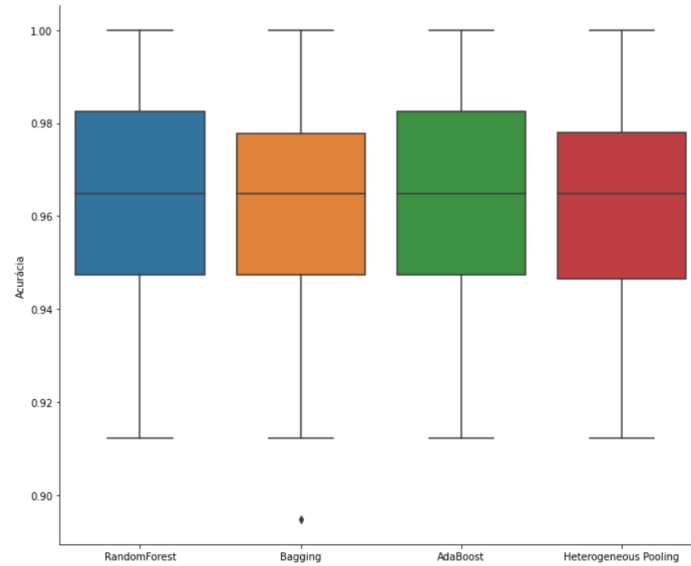


Figure 3: Boxplot obtido para a base breast cancer.

A tabela 7 mostra na matriz triangular superior os resultados do teste t

pareado e na matriz triangular inferior os resultados do teste não paramétrico de wilcoxon.

Bagging	0.007	0.42	0.665
0.007	AdaBoost	0.030	0.028
0.350	0.053	RandomForest	0.873
0.543	0.041	0.882	Heterogeneous Pooling

Table 7: Testes de hipóteses na base breast cancer.

100 Assim como a base *wine*, nem todos os testes rejeitaram a hipótese nula.

4. Conclusões

4.1. Análise geral dos resultados

De maneira geral, os modelos se comportaram de forma semelhante em todas as bases de dados, com exceção do *AdaBoost* quando treinado na base
105 *digits*, onde obteve péssima acurácia. Pode-se perceber também que o classificador *Random Forest* obteve desempenho levemente superior aos demais em todas as bases, exceto na base *breast cancer*, onde o *AdaBoost* obteve melhor desempenho. Além disso, é possível notar que na base *digits* houve diferença significativa entre todos os classificadores, já que todos os testes rejeitaram a
110 hipótese nula. Isso não aconteceu nas bases *wine* e *breast cancer*, pois a maioria dos testes aceitaram a hipótese nula.

4.2. Contribuições do Trabalho

Este trabalho contribuiu para um maior entendimento de técnicas de aprendizagem de máquina aplicadas a problemas de classificação, assim como no
115 entendimento de suas comparações utilizando medidas, gráficos e testes estatísticos.

4.3. Melhorias e trabalhos futuros

Possíveis trabalhos futuros seriam utilizar bases de dados que possam resolver problemas reais.

120 **References**

References

- [1] *Digits, Wine e Breast Cancer* dataset

https://scikit-learn.org/stable/datasets/toy_dataset.html

- [2] Classificadores e *Ensembles*

125 <https://scikit-learn.org/stable/modules/classes.html>

- [3] Testes de hipóteses

<https://docs.scipy.org/doc/scipy/reference/stats.html>