

# Trabalho 1 : Inteligência Artificial

Juliana Camilo Repossi

14 de agosto de 2023

## Resumo

O artigo em questão trata-se de uma comparação experimental entre um conjunto pré-definido de técnicas de aprendizado e classificação automática aplicadas a um problema de classificação. As técnicas escolhidas são: ZeroR (ZR), Bagging (BA), AdaBoost (AB), RandomForest (RF) e Heterogeneous Pooling (HP). Sendo o último um ensemble de natureza heterogênea, que combina os classificadores : Árvore de Decisão (DT), Naive Bayes Gaussiano (NB) e K Vizinhos Mais Próximos (KNN), com o critério de classificação por votação majoritária.

## 1 Introdução

O trabalho aqui detalhado foi proposto pelo Professor Flávio Miguel Varejão, para a disciplina de Inteligência Artificial, ministrada na UFES, e tem como proposta avaliar o desempenho de alguns classificadores vistos em sala de aula e laboratórios da disciplina ministrada. Para o melhor entendimento, o artigo será dividido em subseções. Essas trarão detalhes sobre a base de dados utilizada, sobre o método "Heterogeneous Pooling" implementado, e analisará o desempenho de cada classificador e a possível explicação para cada uma dessas performances.

Por fim, trará uma conclusão geral embasada na análise das métricas obtidas em cada classificador, sendo elas a média das acurácias, seus desvios padrão, intervalos de confiança, e boxplot, além dos valores dos testes t pareado e de wilcoxon para cada par de classificadores. A implementação dos classificadores e das técnicas que serão apresentadas, bem como a geração dos resultados estarão disponíveis no repositório do gitHub em um Jupyter Notebook [disponível nesse link](#) .

Para que os resultados fossem reproduzíveis foram utilizadas 3 tipos de sementes:

- Para o método "RepeatedStratifiedKFold" (Semente = 36851234);
- Para os classificadores com característica aleatória (Semente = 11);
- Para o método "resample" (Semente = acompanhando as iterações) ;

## 2 Base de Dados

### 2.1 Descrição do Domínio

A base de dados usada no trabalho foi obtida de um projeto de pesquisa que visa utilizar informações de imagens de lâmpadas e luminárias de iluminação pública para atualização de cadastro das concessionárias de energia.

## 2.2 Definição das classes e das características

A partir das imagens são extraídas características para compor a base de dados. Três tipos de técnicas foram aplicadas nessa extração: descritores de Fourier, descritores de Hu e descritores de Haralick. A base completa obtida contém 10 descritores de Fourier, 7 descritores de Hu e 6 descritores de Haralick, totalizando 23 características.

Já as classes são caracterizadas pelo tipo: Lâmpada de vapor de mercúrio (Mercury Vapor - MV), lâmpada de vapor de sódio de alta pressão (High Pressure Sodium - HPS) e lâmpada de vapor metálico (Metal Halide - MH) e pela potência: 70, 100, 125, 150, 250 ou 400W da lâmpada. A Tabela 1 mostra a quantidade e distribuição de exemplos por classe. Nota-se que a base é razoavelmente balanceada.

<b>Tabela 1 – Distribuição das exemplos por classe</b>				
Classe	Tipo	Potência (W)	Quantidade	(%)
HPS070	HPS	70	30	10,1%
HPS100	HPS	100	32	10,8%
HPS150	HPS	150	35	11,8%
HPS250	HPS	250	33	11,1%
HPS400	HPS	400	37	12,5%
MH150	MH	150	23	7,7%
MH250	MH	250	49	16,5%
MH400	MH	400	37	12,5%
MV125	MV	125	21	7,1%
Total			297	100,0%

Figura 1: Distribuição de classes da base

## 2.3 Número de instâncias

Essa abordagem utilizou-se da capacidade total da base de dados supracitada. Sendo esta composta pelas 23 características de cada uma das 297 instâncias de lâmpadas e luminárias. Logo temos uma entrada de 297 linhas por 23 colunas.

## 2.4 Tratamento dos dados

O tratamento feito aos dados foi extremamente simplório. Após o carregamento, foi preciso apenas fazer a junção das colunas "tipo.lampada" e "potencia" que vinham separadas da base de dados. A junção se justifica pelo fato de que a combinação dessas características representará uma única classificação no nosso domínio. Esta será a lista de classificações utilizada. Também foram extraídas da base todas as 23 características produzidas pelos descritores mencionados, nelas serão feitas as devidas adaptações de representação de um número real (com a troca de ',' para '.' para o padrão do python).

## 3 O método Heterogeneous Pooling

O classificador Heterogeneous Pooling implementado é ensemble heterogêneo que usa como classificadores base: Árvore de Decisão (DT), Naive Bayes Gaussiano (NB) e K Vizinhos Mais Próximos (KNN), sempre com valores default do sklearn para seus hiperparâmetros. Os únicos parâmetros do método Heterogeneous Pooling são a semente para a construção do classificador DT e o "n\_samples", que indica o número de vezes que cada um desses classificadores será criado. Por exemplo, se "n\_samples" é igual a 3, o combinado será composto por 9 classificadores, dentre eles: 3 árvores de decisão, 3 naive bayes e 3 vizinhos mais próximos.

Para diferenciar os classificadores de mesmo tipo em um combinado, o primeiro deles será treinado com a base de treino original e os demais serão treinados com uma base de treino diferente, obtida a partir da base de treino original através da função `resample` do `sklearn`, que utilizará como semente o número da iteração, indo de 0 a `n_samples- 1`.

O critério de decisão para classificar uma instância será por votação majoritária, ou seja, deve-se escolher a classe mais escolhida dentre os classificadores que compõem o combinado. Em caso de empate, a classe escolhida deve ser a mais frequente na base de dados de treino original dentre as que empataram na votação.

## 4 Descrição dos Experimentos Realizados e seus Resultados

Para essa seção, será realizada uma avaliação do desempenho real de cada classificador frente ao que era esperado na teoria. Para isso, teremos como apoio a tabela a seguir que compila os dados de desempenho de cada um dos classificadores frente as estratégias de validação cruzada estratificada e busca em grade.

	Média	Desvio Padrão	Limite Inferior	Limite Superior
<b>ZR</b>	0.165057	0.010883	0.161163	0.168952
<b>BA</b>	0.635632	0.074520	0.608966	0.662298
<b>AB</b>	0.339042	0.029170	0.328604	0.349480
<b>RF</b>	0.623142	0.074574	0.596456	0.649827
<b>HP</b>	0.581418	0.090653	0.548979	0.613857

Figura 2: Métricas das acurácias de cada classificador

Considerações:

- Primeiro Classificador : ZeroR  
Como o esperado o zeroR teve um desempenho muito ruim, já que é apenas um classificador base-line. Esse mal desempenho se evidencia ainda mais, nesse caso, pelo fato do problema ter uma base balanceada de classes, como visto na figura 1, que representa uma desvantagem para seu modelo de implementação.
- Segundo Classificador : Bagging  
O bagging, um ensemble homogêneo que trabalha em paralelo, por padrão com uma `DecisionTreeClassifier`, já obtém resultados bem mais otimistas, e isso pode ser consequência de uma base de dados mais instável, no qual pequenas modificações na entrada de dados resultam em grandes diferenças na classificação. Esse perfil pode favorecer o bagging e prejudicar um classificador KNN, por exemplo.
- Terceiro Classificador : AdaBoost  
O AdaBoosting apesar de também trabalhar com a `DecisionTreeClassifier` é um ensemble homogêneo que trabalha em série. O que não se revelou tão vantajoso se comparado ao Bagging. Isso pode ser observado pela acurácia do modelo ter atingido apenas próximo da metade da supracitada.
- Quarto classificador : RandomForest  
Com um bom desempenho também, o RandomForest vem com a sua estratégia de seleção de instâncias (linhas) aleatórias na base, assim como das características para construir cada uma das árvores. E isso traz um bom desempenho, provavelmente pelo mesmo motivo já discutido no bagging, já que são metodologias próximas de trabalho.
- Quinto classificador : Heterogeneous Pooling  
Para finalizar, o quinto classificador, teve um desempenho intermediário se comparado aos demais.

Podemos supor que essa pequena diferença em relação aos melhores possa estar atrelada ao fato deste usar como um de seus três classificadores o KNN. Essa proposição se justifica devido ao Bagging e o RandomForest terem tido uma boa performance, assim temos uma grande chance de estarmos trabalhando com dados instáveis e, o KNN sem hiperparâmetros pode não colaborar tanto para cobrir uma maior área de acerto de predições, nos quais a Árvore de Decisão (DT) e Naive Bayes Gaussiano (NB) já não contemplem. Já que o primeiro teve bom desempenho no ensemble paralelo, e o segundo se utiliza de uma base estatística para prever a classificação final. Assim, o HP fica em segunda escolha frente aos dois mais performáticos dessa pesquisa.

Ainda com a pretensão de trazer embasamento para a escolha de um classificador mais compatível com o problema dado, temos a imagem 3, no qual foram feitos boxplots das acurácias de cada um dos classificadores citados.

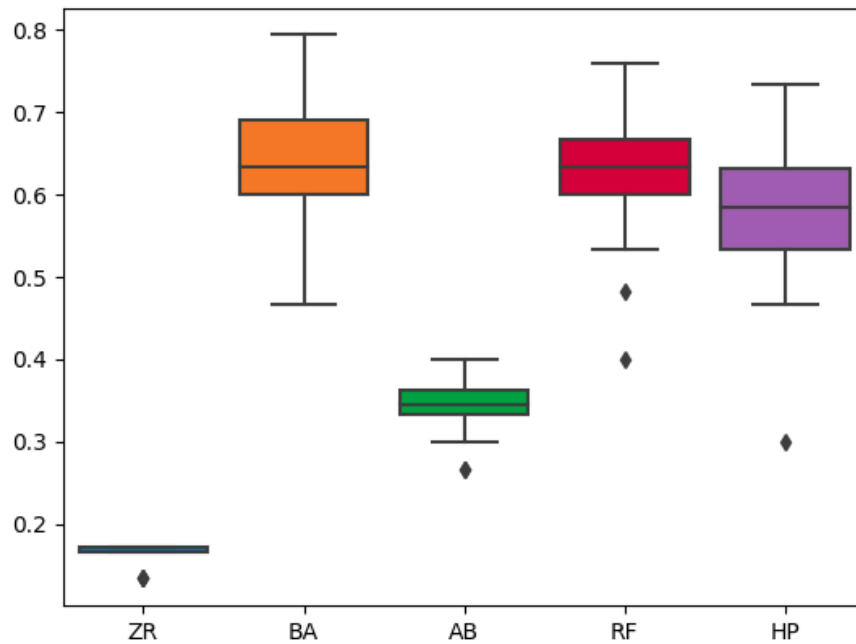


Figura 3: Boxplot das acurácias dos classificadores

Com essa representação visual das acurácias, reafirmamos o desempenho precário do ZeroR e do AdaBoost, frente a esse problema e, exaltamos boas performances para os três demais. Aqui podemos destacar mais algumas características como:

- O heterogeneus não seria a primeira escolha para este problema, pois ele possui um outlier bem abaixo dos três demais e aproximadamente 75% das suas acurácias estão abaixo de 0.65, enquanto no BA e RF, apenas 50% estão com acurácias abaixo desse valor.
- Entre o RF e o BA há poucas diferenças, sendo elas:
  - A metade superior do Bagging consegue alcançar maiores acurácias se comparado ao Random Forest;
  - Já os valores 25% inferiores alcançam valores menores no BA, se comparados aos do RF, porém este apresenta 2 outliers;

Por fim, foi proposto realizar o teste t pareado e de wilcoxon entre cada par de resultados dos classificadores, sendo que valores maiores ou iguais a 0,05 são traduzidos em, não há diferenças estatísticas entre os dois classificadores avaliados a 95% de significância.

Mais uma vez podemos reiterar as hipóteses anteriormente levantadas sobre o desempenho otimista e próximo que o Bagging e o RandomForest apresentaram. Tanto no primeiro teste quanto no segundo, eles

ZR	<b>0.0</b>	<b>0.0</b>	<b>0.0</b>	<b>0.0</b>
<b>0.000002</b>	BA	<b>0.0</b>	0.44258	<b>0.001445</b>
<b>0.000001</b>	<b>0.000002</b>	AB	<b>0.0</b>	<b>0.0</b>
<b>0.000002</b>	0.593239	<b>0.000002</b>	RF	<b>0.011317</b>
<b>0.000002</b>	<b>0.002907</b>	<b>0.000002</b>	<b>0.017078</b>	HP

obtiveram valores maiores que 0,05 apontando que não há diferenças significativas entre eles a um nível de significância de 95%. Todos os demais tem diferenças significativas entre si.

## 5 Conclusão

### 5.1 Análise geral dos resultados

Com as evidências supracitadas fica fácil concluir que o Bagging e o RandomForest tiveram mais performance nesse estudo, isso pode ser justificado pois esses métodos tem algum tipo de semelhança em suas implementações: os dois possuem a árvore de decisão como classificador base, votação majoritária para escolher o resultado, e utilizam-se de diferentes conjuntos para treinar em paralelo o mesmo classificador. Essa pode ser uma boa justificativa para os dois não apresentarem diferenças significativas nesse contexto. O desempate poderia considerar o tempo de execução entre os dois, por exemplo, se isso fosse importante para o cliente.

Por outro lado, o Heterogeneous Pooling atinge um desempenho um pouco inferior aos dois melhores, porém os testes t e de Wilcoxon rejeitam a hipótese nula, então apesar de resultados próximos nas métricas, eles tem diferenças significativas entre si, o que pode ser visualizado na análise do boxplot feita. E o zeroR e o AdaBoost tem resultados péssimos, sendo assim, não devem ser ao menos cogitados para esse caso.

### 5.2 Contribuições do Trabalho

Esse tipo de projeto escalaria a classificação de lâmpadas e luminárias de iluminação pública para a atualização de cadastro das concessionárias de energia, visto que um processo manual exigiria uma grande equipe e um longo tempo para finalizar a tarefa. Considerando ainda que, se tratando de um domínio específico, essa equipe precisaria de treinamento para reconhecer as lâmpadas, o que estende ainda mais o tempo e o custo da operação. A Inteligência Artificial vem como grande aliada nesse caso.

### 5.3 Melhorias e trabalhos futuros

Do ponto de vista dessa análise, essa seria a proposição inicial para as concessionárias, podendo ser desenvolvidos mais estudos envolvendo outros classificadores que respondem bem frente às características da base já inferidas aqui. Poderiam também ser realizados testes com hiperparâmetros diferentes, testar se um subconjuntos diferentes de colunas poderiam ter melhores resultados. E, até mesmo, em uma abordagem mais complexa, que exigiria um maior conhecimento dos descritores utilizados, testar se paralelizar a classificação em : um conjunto de características e técnicas para prever o tipo da lâmpada e um conjunto de características e técnicas para prever a potência da lâmpada e depois concatená-los traria melhor desempenho para o domínio, ou não.

## 6 Referências Bibliográficas

As referências para a escrita desse relatório foram obtidas a partir dos materiais de aula disponibilizados pelo Dr. Flávio Miguel Varejão, bem como suas aulas expositivas e laboratórios ministrados na disciplina de Inteligência Artificial - UFES.