

# Trabalho de Inteligência Artificial

Dhiego Santos Broetto

*Universidade Federal do Espírito Santo*

*Ciência da Computação*

*Outubro 2019*

---

## **Abstract**

Este trabalho consiste em realizar uma comparação experimental entre um conjunto predefinido de técnicas de aprendizado e classificação automática aplicadas a alguns problemas de classificação. As técnicas escolhidas são: ZeroR, OneR, OneR Probabilístico, Centróide, Centróide OneR, Naive Bayes Gaussiano, Knn, Árvore de Decisão, Rede Neural e Florestas de Árvores. As bases de dados a serem utilizadas são Iris, Digits, Wine e Breast Cancer.

---

## 1. Introdução

Com o intuito de aprendizagem, neste presente trabalho foi feita uma comparação experimental entre um conjunto de técnicas de aprendizado e classificação automática aplicadas a alguns problemas de classificação, utilizando determinadas bases de dados.

Essa comparação foi dividida em duas etapas, a primeira etapa consiste em treinamento e teste com validação cruzada de 10 *folds* dos classificadores que não possuem hiperparâmetros, são eles: ZeroR, OneR, OneR Probabilístico, Centróide, Centróide OneR e Naive Bayes Gaussiano (GaussianNB).

Já na segunda etapa o processo é dado pelo treinamento, validação e teste dos classificadores que precisam de ajuste de hiperparâmetros, que são eles: KNN, Árvore de Decisão (DecisionTreeClassifier), Redes Neurais (MLPClassifier) e Florestas de Árvores (RandomForestClassifier).

O trabalho contém tabelas com média das acurácias e desvio padrão dos resultados obtidos em cada *fold* do ciclo externo de cada classificador. Além de haver boxplots dos resultados de cada classificador em cada *fold*. Também apresenta uma tabela comparativa entre a média de todos os resultados de acurácia e desvio padrão de todos os classificadores, considerando as 4 bases junto de gráficos em barra destes valores.

## 2. Descrição Resumida dos Datasets

Dentro do *scikit-learn*, existem diversos datasets para se realizar experimento e treinar classificadores. Dentre eles, foram selecionadas e utilizadas quatro bases para trabalhar nas duas etapas. As bases escolhidas são:

### 2.1. Iris

Essa talvez seja a base de dados mais conhecida encontrada na literatura de reconhecimento de padrões. O conjunto de dados contém 3 classes de 50 instâncias cada, em que cada classe se refere a um tipo de planta de íris.

## 2.2. *Digits*

O conjunto de dados contém imagens de dígitos manuscritos: 10 classes em que cada classe se refere a um dígito. Os programas de pré-processamento disponibilizados pelo NIST foram usados para extrair bitmaps normalizados de dígitos manuscritos de um formulário pré-impresso. Os bitmaps de 32x32 são divididos em blocos sem sobreposição de 4x4 e o número de pixels em são contados em cada bloco. Isso gera uma matriz de entrada de 8x8, em que cada elemento é um número inteiro no intervalo de 0 a 16. Isso reduz a dimensionalidade e dá invariância a pequenas distorções.

## 2.3. *Wine*

Esses dados são os resultados de uma análise química de vinhos cultivados na mesma região da Itália, mas derivados de três cultivares diferentes. A análise determinou as quantidades de 13 constituintes encontrados em cada um dos três tipos de vinhos.

## 2.4. *Breast Cancer*

Um método de classificação que utiliza programação linear para construir uma árvore de decisão. As características relevantes foram selecionadas usando uma pesquisa exaustiva no espaço de 1-4 características e 1-3 planos de separação.

Para mais detalhes das bases podem ser encontradas na seção 6 de referências bibliográficas.

# 3. Descrição Resumidas dos Métodos Implementados

No trabalho, alguns classificadores já estão implementados através do *scikit-learn*, porém alguns precisaram ser desenvolvidos para efeito de comparação com os demais. São estes:

### 3.1. ZeroR

O ZeroR é o classificador mais simples para se implementar do que os demais escolhidos. Ele seleciona apenas o rótulo que possui o maior número de ocorrências, considerando todas as instâncias. Assim, classifica todas as instâncias com esse rótulo selecionado no treinamento.

### 3.2. OneR

OneR é um classificador que utiliza valores discretizados e uma tabela de contingência para obter resultados mais satisfatórios. Primeiramente as características são todas discretizadas pelo número de rótulos e intervalos de mesma instância. Após isso é montada uma tabela de contingência, na qual se contabiliza o total de ocorrências daquele valor para cada rótulo para todos os valores das instâncias. Próximo passo é escolher como melhor atributo aquele que possui a maior acurácia através da comparação com os demais e selecionar quais rótulos, selecionando o que obtiver a maior quantidade de ocorrências de um rótulo em cada instância daquele atributo.

### 3.3. OneR Probabilístico

Diferentemente do que ocorre no OneR, o probabilístico atua no momento em que se classifica uma instância de rótulo desconhecido. Aqui ele faz a utilização do método da roleta, que seleciona um dos rótulos através de um peso correspondente, este dado pelo valor dividido pelo valor da soma de todos os rótulos.

### 3.4. Centróide

O classificador baseado no centróide funciona da seguinte forma: agrupa as instâncias de um mesmo rótulo, encontra uma instância média entre eles e depois na predição, calcula a menor distância entre os atributos da instância da previsão e a média de um determinado rótulo. Daí seleciona o que obtiver a menor distância para classificar cada instância.

### 3.5. *Centróide OneR*

Basicamente a junção do OneR com o Centróide. Aqui se aplica o OneR para a seleção do melhor atributo, dado isso, se agrupa os valores não discretizados que possuem os mesmos rótulos do atributo selecionado da base de treino. A partir daí se usa o centróide para encontrar um ponto médio em cada agrupamento e definir a menor distância entre esses pontos encontrados e os atributos das dadas instâncias do teste.

## 4. Descrição dos Experimentos Realizados

Os experimentos foram divididos em duas etapas, primeira etapa no treinamento e teste dos classificadores que não possuem hiperparâmetros e na segunda os que possuem. Portanto, a seção será dividida em duas subseções para tratar cada etapa separadamente.

### 4.1. *Primeira Etapa*

Para início dos experimentos, foram escolhidas quatro bases (datasets) conhecidas no meio acadêmico e 6 classificadores que não possuem hiperparâmetros, 5 que foram implementados, sendo eles ZeroR, OneR, OneR Probabilístico, Centróide, Centróide OneR conforme descritos na seção 3 de métodos implementados e o Naive Bayes Gaussiano (GaussianNB), que já está incluso no *scikit-learn*.

Para cada dataset, foram feitos treinamentos e testes com validação cruzada de 10 *folds*. A partir daí foi gerada uma tabela com os valores da média e desvio padrão de cada classificador, além de boxplots com scores para comparação entre os classificadores utilizados.

Nas próximas subseções serão apresentados e analisados os resultados obtidos em cada dataset.

#### 4.1.1. *Iris*

Na base Iris, podemos observar que os resultados das médias de cada classificador são bem parecidas, com exceção do ZeroR. Mas quem saiu na frente foi o Centroid OneR, com em torno de 96,6%. O Probabilistic OneR poderia ter

se saído melhor, já que ele se utiliza da roleta, que deixa o resultado aleatório, também vale ressaltar *outlier* bem fora do resultado no boxplot, que desvaloriza um pouco a confiança no método. Outra coisa a se observar é que apesar do resultado do ZeroR, seu desvio padrão é quase 0, isso implica que ele é o classificador mais confiável em quesito de variância, porém é de longe o melhor em apresentar um resultado satisfatório.

Classificador	Média	DP
ZeroR	0.333333333333337	<b>5.551115123125783e-17</b>
OneR	0.9533333333333334	0.06699917080747259
Probabilistic OneR	0.8399999999999999	0.155492050529208
Centroid	0.9333333333333333	0.059628479399994376
Centroid OneR	<b>0.9600000000000002</b>	0.0533333333333332
GaussianNB	0.9533333333333334	0.04268749491621898

Table 1: Tabela Iris da primeira etapa

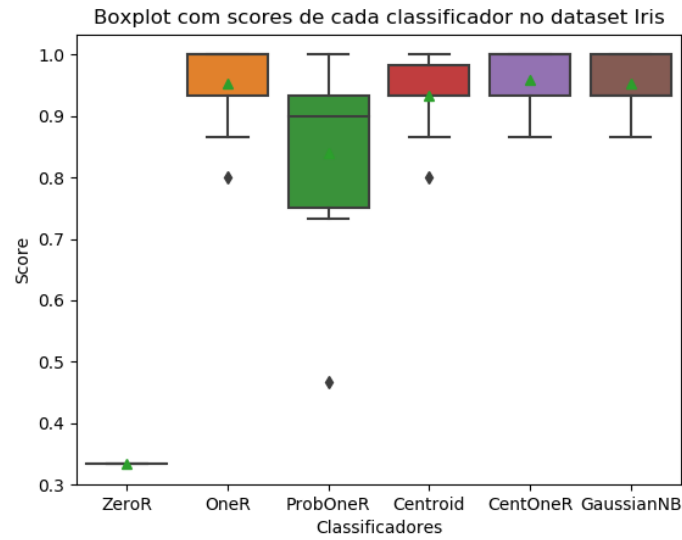


Figure 1: Iris

#### 4.1.2. Digits

Para Digits o cenário muda um pouco. Se observa que os valores das medições ficaram bem dispersas e que a maioria apresentou o valor das médias bem abaixo do que os da base Iris. Dá pra considerar também que todos os classificadores apresentaram desvios padrão bem pequenos, que é ótimo. Mas apesar da maioria apresentar baixos valores dos scores, o classificador baseado no Centróide apresentou o melhor valor, com aproximadamente 88,3%, seguido do GaussianNB que obteve em torno de 81%. Aqui também observamos que o desvio padrão do ZeroR foi o menor entre todos, se destacando mais uma vez apenas no desvio padrão com valor aproximado de 0,1%.

Classificador	Média	DP
ZeroR	0.10127425688130931	<b>0.001274401867253491</b>
OneR	0.23303354606521257	0.021685445382370397
Probabilistic OneR	0.17484969814656162	0.027697560953875346
Centroid	<b>0.8836101717889818</b>	0.04112678822301056
Centroid OneR	0.23363618614405857	0.024133635663880226
GaussianNB	0.8103537583567821	0.056655402070708565

Table 2: Tabela Digits da primeira etapa

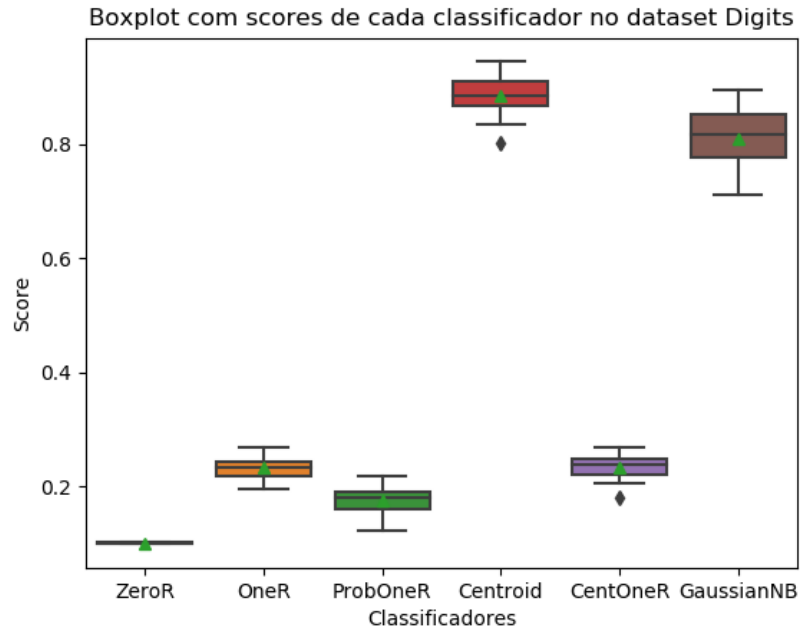


Figure 2: Digits

#### 4.1.3. Wine

As medições na base Wine já possui valores mais interessantes. De cara já dá pra destacar que o GaussianNB obteve o melhor resultado médio de acurácia com aproximadamente 96,1% e o ZeroR com o melhor desvio padrão aproximado de 1,6%, mais uma vez. Aqui também vemos que tanto o Centróide quanto o Centróide OneR quase não se diferenciaram, com acurácias em torno de 72,1% e 73,2% e desvio padrão 8,49% e 8,42% respectivamente. Já o OneR foi o que mais apresentou *outliers*, que comprometem um pouco a confiabilidade do classificador. Mais uma vez o OneR Probabilístico saiu atrás por conta de sua aleatoriedade com score em torno de 56,3%.



Classificador	Média	DP
ZeroR	0.3992539559683522	<b>0.016871583549832498</b>
OneR	0.6235208118335054	0.07712192798491828
Probabilistic OneR	0.5633212934296525	0.09153801714089298
Centroid	0.7216073271413829	0.08490131985948657
Centroid OneR	0.7327184382524939	0.0842522501710735
GaussianNB	<b>0.9616959064327485</b>	0.042442001415448946

Table 3: Tabela Wine da primeira etapa

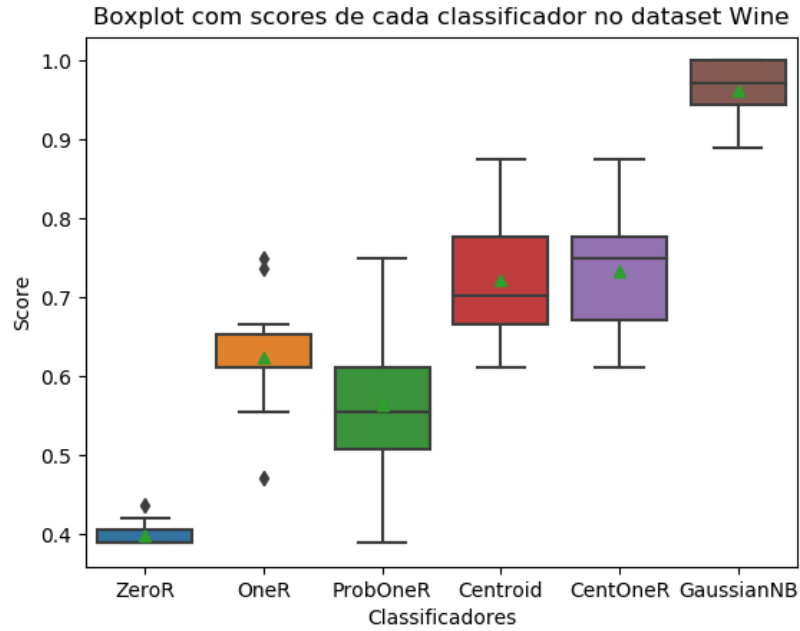


Figure 3: Wine

#### 4.1.4. Breast Cancer

Com o Breast Cancer, com exceção do ZeroR, todos apresentaram resultados satisfatórios nos scores. Mas quem apresentou o melhor resultado foi novamente o GaussianNB com aproximadamente 93,8% e mais uma vez o ZeroR no desvio padrão em torno dos 0,44%. Apesar de que o OneR Probabilístico obteve um

resultado aceitável, ainda não foi bom o suficiente em comparação com os demais. Vale frizar que independentemente do ótimo desempenho do GaussianNB, ainda apresentou *outlier*.

Classificador	Média	DP
ZeroR	0.6274274047186933	<b>0.00441189245975379</b>
OneR	0.9141463572724915	0.039822475517287674
Probabilistic OneR	0.8594751966122202	0.03200331678117153
Centroid	0.8913641863278887	0.03879383921566746
Centroid OneR	0.9034309912712816	0.036006020163806926
GaussianNB	<b>0.9386796733212339</b>	0.030112887483699426

Table 4: Tabela Breast Cancer da primeira etapa

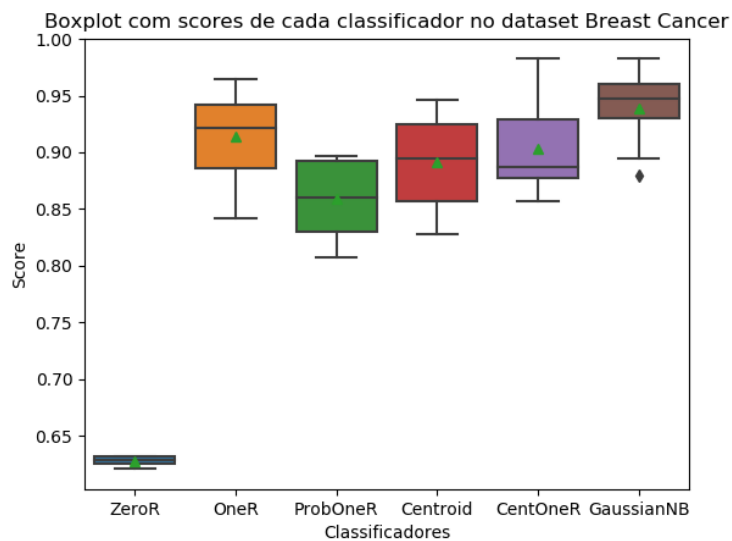


Figure 4: Breast Cancer

#### 4.2. Segunda Etapa

Na segunda etapa, foram escolhidos para a realização dos experimentos os classificadores KNN, Árvore de Decisão, Rede Neural e Florestas de Árvores, cada um com seus respectivos conjunto de hiperparâmetros, conforme descritos na tabela 5 abaixo.

Classificador	Hiperparâmetros
K Nearest Neighbor (KNN)	'n_neighbors': [1,3,5,7,10]
Árvore de Decisão (DTC)	'max_depth': [None,3,5,10]
Rede Neural (MLPC)	'max_iter': [50,100,200], 'hidden_layer_sizes': [(15,)]
Florestas de Árvores (RFC)	'n_estimators': [10,20,50,100]

Table 5: Tabela de hiperparâmetros dos classificadores

Para cada dataset, foram feitos treinamentos, validações e testes através de dois ciclos aninhados, um interno com *grid search* de 4 *folds* e outro externo de validação cruzada com 10 *folds*. A partir daí gerados uma tabela cada com o valor da média e desvio padrão, além de boxplots com scores para comparação dos classificadores.

##### 4.2.1. Iris

No dataset Iris, podemos ver que o classificador Árvore de Decisão obteve o maior score médio e melhor desvio padrão do que os demais, mas por pouco também. Em contrapartida, o classificador Rede Neural foi o que apresentou o pior desempenho, com acurácia média de aproximadamente 71%, dado que o número de iterações no *grid search* foram abaixo do esperado para que o Rede Neural alcançasse sua convergência no treinamento dos pesos, isso pode ser notado ao se executar a segunda etapa, o *scikit-learn* disparou diversos *ConvergenceWarnings* no decorrer da execução. Na execução da segunda etapa, com exceção do Rede Neural, todos apresentaram acurácia acima dos 95% e desvios padrão abaixo dos 6%, em comparação com a primeira etapa em que houveram scores acima dos 83% e abaixo de 16% de desvios padrão.

Classificador	Média	DP
KNN	0.9533333333333334	0.059999999999999984
DTC	<b>0.9666666666666666</b>	<b>0.03333333333333326</b>
MLPC	0.7133333333333334	0.20666666666666667
RFC	0.96	0.044221663871405324

Table 6: Tabela Iris da segunda etapa

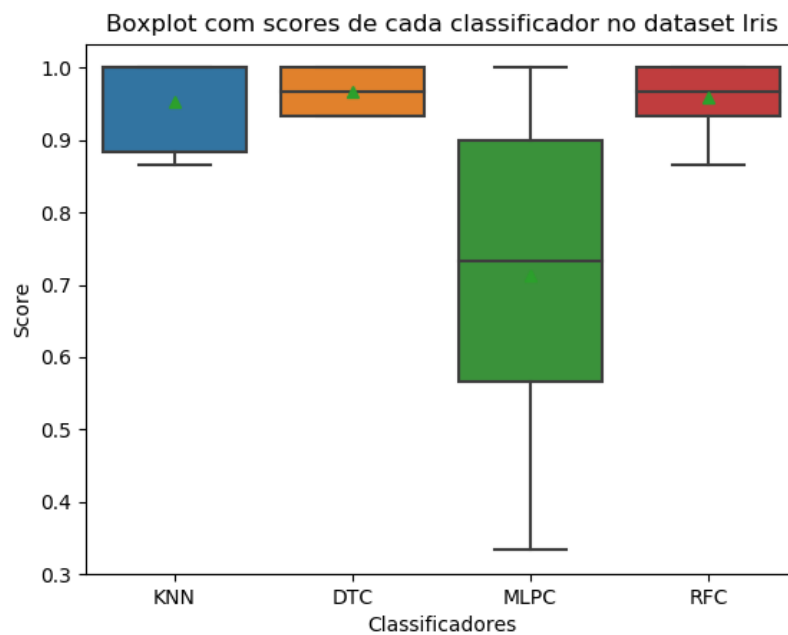


Figure 5: Iris

#### 4.2.2. Digits

Já na base Digits, podemos observar o contrário, apesar do baixo número de iterações no *grid search* que impacta no classificador de Rede Neural, ainda conseguiu obter um resultado médio melhor do que o da Árvore de Decisão. Mas quem obteve a melhor acurácia foi o KNN com aproximadamente 97,8% de acurácia e melhor desvio padrão com aproximadamente 1,7%. Comparando com a primeira etapa, que obteve resultados bem dispersos e baixos, na segunda

etapa já foram acurácias acima dos 82%.

Classificador	Média	DP
KNN	<b>0.9788938332167104</b>	<b>0.017601243897194148</b>
DTC	0.8232394270536018	0.03249967328845698
MLPC	0.9226871527637905	0.04196831060356903
RFC	0.9555541883709425	0.021147489474344968

Table 7: Tabela Digits da segunda etapa

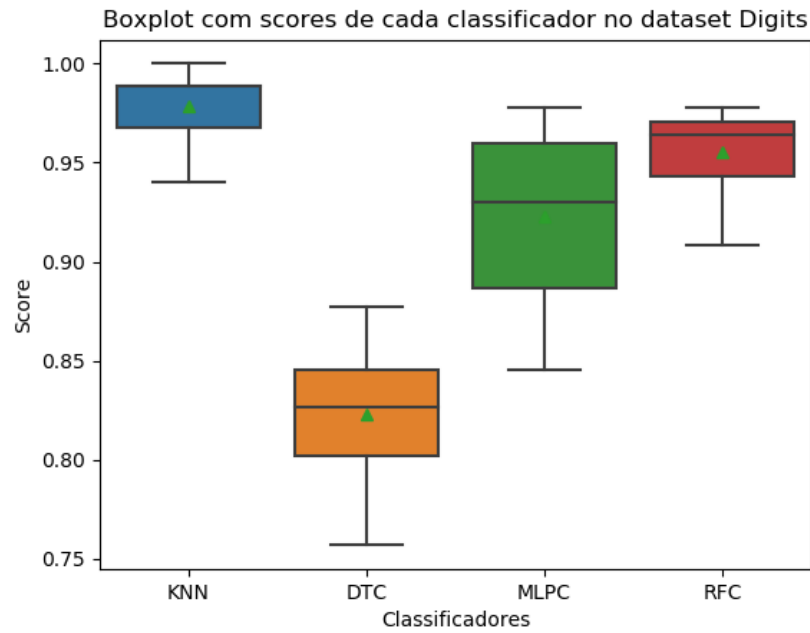


Figure 6: Digits

#### 4.2.3. Wine

Numa primeira visão das medições sob a base Wine, destacamos o problema do baixo número de iterações também afetou o desempenho do classificador de Rede Neural que ficou com score médio de aproximadamente 51,3%. O único classificador que entregou uma acurácia acima dos 90% foi o de Florestas de Árvores, com aproximadamente 97,1% e também apresentou o melhor desvio padrão com aproximadamente 2,8%. Podemos dizer que os resultados em geral não foram tão diferentes nos experimentos da primeira e segunda etapa em comparação com os das outras bases, mas vale destacar que os desvios padrão da primeira etapa tiveram melhores valores (abaixo dos 9,2%) do que os da segunda (abaixo dos 19,6%).

Classificador	Média	DP
KNN	0.7089245786033712	0.12428744272161676
DTC	0.8891468868249055	0.0783642134115304
MLPC	0.5133621431028552	0.19558715241796218
RFC	<b>0.9718954248366012</b>	<b>0.028119770933724135</b>

Table 8: Tabela Wine da segunda etapa

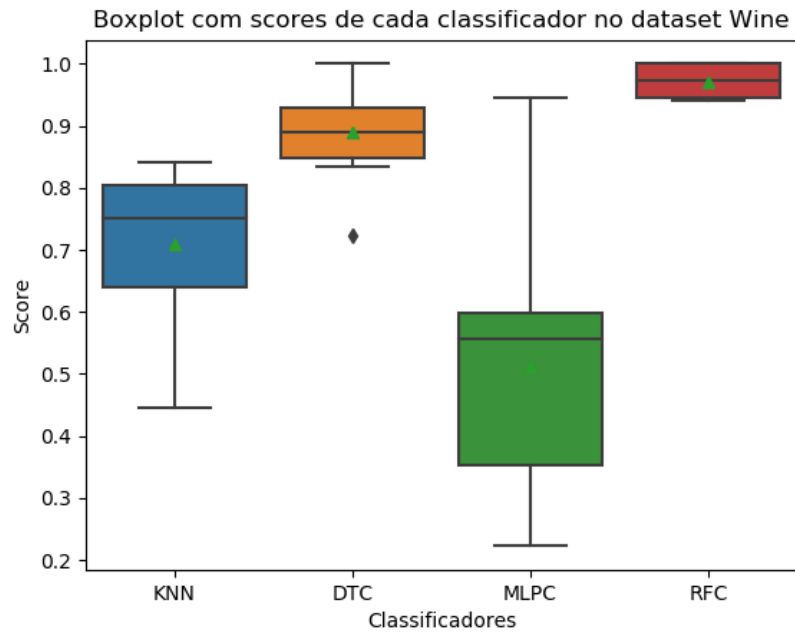


Figure 7: Wine

#### 4.2.4. Breast Cancer

Como ocorrido na base Iris e na base Wine, na base Breast Cancer destacamos o problema de que classificador de Rede Neural mais uma vez foi afetado pelo baixo número de iterações, sofreu tanto na acurácia quanto no desvio padrão, além de apresentar *outliers* bem comprometedores. Já quem saiu na frente foi o classificador de Floresta de Árvores em ambos os casos, acurácia e desvio padrão. Mas mesmo assim, os scores ficaram mais equilibrados, com médias acima dos 91,7% (tirando o Rede Neural), em comparação com a primeira etapa que obteve médias acima de 85,9% (desconsiderando o ZeroR).

Classificador	Média	DP
KNN	0.9316588886008124	0.031245167972901115
DTC	0.9176216403076658	0.046717364496935934
MLPC	0.7249827154092127	0.2665550974529333
RFC	<b>0.9597042174401521</b>	<b>0.027947022081983387</b>

Table 9: Tabela Breast Cancer da segunda etapa

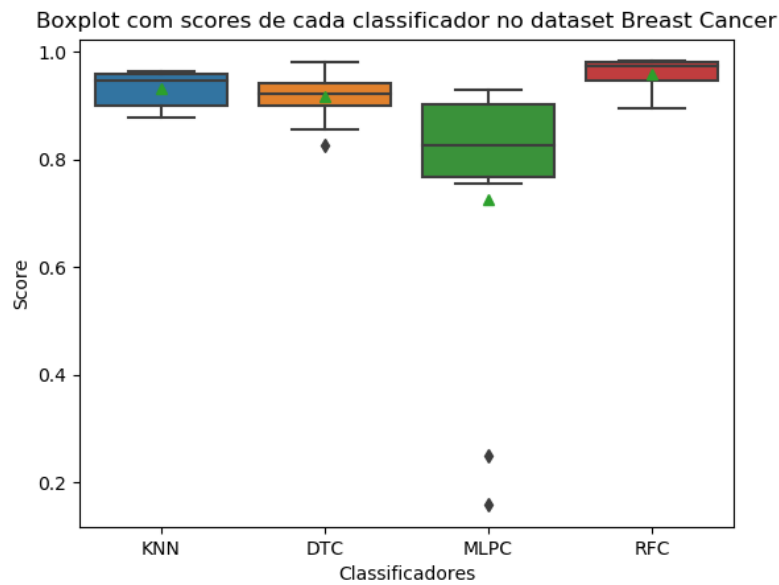


Figure 8: Breast Cancer

## 5. Conclusões

Como proposto, este trabalho foi elaborado para fins de estudo de vários classificadores de *Machine Learning* para classificar bases (datasets) predeterminados. Os experimentos foram divididos em duas etapas, primeira etapa para execução dos classificadores *baselines* que, não possuem ajustes de hiperparâmetros e foram submetidos à uma validação cruzada de 10 *folds*. Já os da segunda etapa, que possuem ajustes de hiperparâmetros, foram executados em duas iterações aninhadas, um interno que faz um *grid search* de 4 *folds* e outro



externo de validação cruzada com 10 *folds*.

### 5.1. Análise geral dos resultados

Como análise geral, vale destacar o desempenho do GaussianNB e do Floresta de Árvores, que possuíram os melhores scores dentre a primeira etapa e a segunda respectivamente, com valores em torno de 91,6% e 96,1%, conforme dados das médias dos resultados em todas as bases na tabela 10 e gráficos 9 e 10.

Outro ponto importante foi que nenhum classificador ficou abaixo do resultado apresentado pelo ZeroR, já que ele é muito utilizado como fator comparativo. Se um determinado classificador apresentar resultados abaixo do medido pelo ZeroR, pode descartar tal classificador.

Em média o OneR Probabilístico ficou um pouco abaixo do OneR, por outro lado o Centróide OneR obteve resultados mais satisfatórios do que o OneR, mas perde para o Centróide. Dados resultados poderiam ter sido diferentes, já que o OneR Probabilístico é aleatório e apresenta scores diferentes a cada execução.

O classificador mais prejudicado foi o de Rede Neural, que ficou bem abaixo da expectativa, devido aos baixos números de iterações. Problema este que foi disparado pelo *scikit* como forma de *ConvergenceWarnings* durante os experimentos da segunda etapa.

Em geral, os classificadores da segunda etapa acertaram em média 25,9% a mais em comparação aos da primeira etapa, porém obtiveram 66,9% mais de desvio padrão do que os da primeira etapa. Logo podemos concluir que os classificadores da segunda etapa atingiram melhores resultados em acurácia, porém os *baselines* são quem possuíram menores desvios padrão.

	Classificador	Média	DP
Primeira Etapa	ZeroR	0.3653222377	<b>0.005639469469</b>
	OneR	0.6810085121	0.05140725492
	Probabilistic OneR	0.609411547	0.07668273635
	Centroid	0.8574787546	0.05611260667
	Centroid OneR	0.7074464039	0.04943130983
	GaussianNB	<b>0.9160156679</b>	0.04297444647
	Média total	0.6894471872	0.04704130395
Segunda Etapa	KNN	0.8932026584	0.05828346365
	DTC	0.8991686552	0.04772864613
	MLPC	0.7185913362	0.1776943068
	RFC	<b>0.9617884577</b>	<b>0.03035898659</b>
	Média total	0.8681877769	0.07851635079

Table 10: Tabela com as médias dos scores e desvios padrão

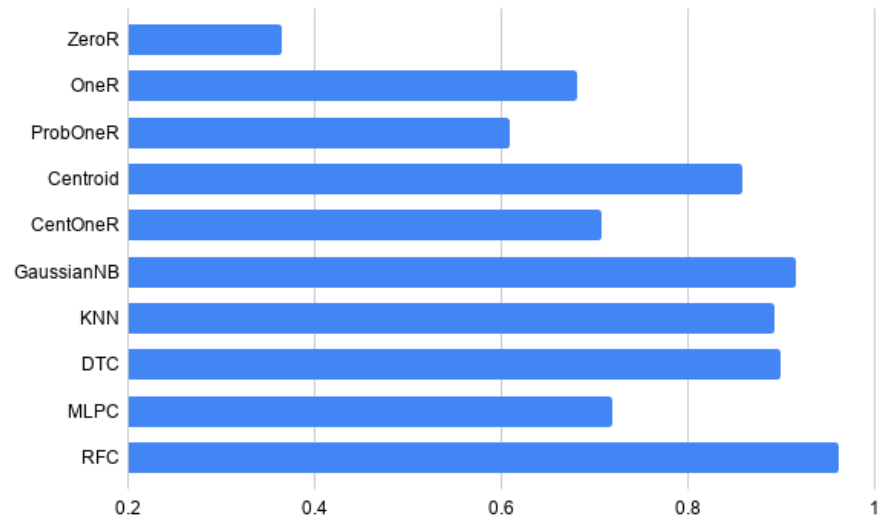


Figure 9: Gráfico das médias dos scores

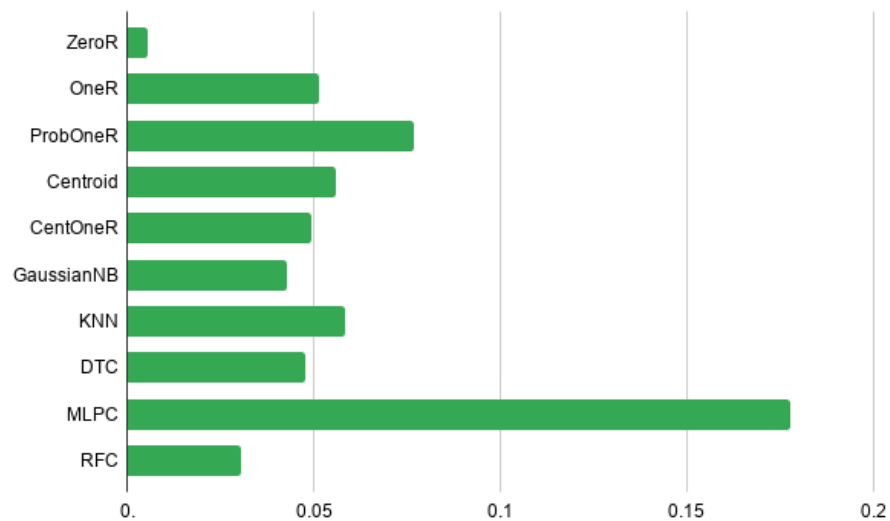


Figure 10: Gráfico das médias dos desvios padrão

### 5.2. Contribuições do Trabalho

O trabalho serviu para entender como que funciona e são executados métodos de classificação de bases através de treinamento e teste de *machine learning*.

Um outro ponto pode ser explorado através da escolha de diferentes formas de executar o experimento, pois dependendo das estratégias adotadas, um classificador dito como bom acaba perdendo sua eficiência.

Como fonte de aprendizado, a comparação de resultados obtidos através de diversos classificadores ajuda na percepção e possíveis ajustes que podem ser feitos futuramente.

### 5.3. Melhorias e trabalhos futuros

Tendo em vista os resultados, seria fundamental aumentar a quantidade de iterações para ver o funcionamento dos classificadores e comparar com o presente trabalho, para ver se há melhoras ou pioras. Também pode-se mexer nos ajustes de hiperparâmetros, já que no caso o classificador de Rede Neural foi prejudicado. Caso sejam feitos esses ajustes, o MLPC poderia apresentar resultados mais interessantes.

Um aumento no número de bases (datasets) utilizados também seria bom para ver em quais bases os classificadores mais possuem facilidades e dificuldades de encontrar uma acurácia boa.

Utilização de mais classificadores diferentes para fins comparativos, principalmente na segunda etapa, já que este contou apenas com 4 classificadores. Talvez teria uma comparação mais balanceada, olhando pra primeira etapa que houveram 6 classificadores.

Também pode ser considerado a execução de várias vezes um dado cenário para um mesmo classificador e calcular a média desses resultados, já que quando se executa uma única vez um classificador que possui um algoritmo aleatório, como o OneR Probabilístico, pode acontecer de gerar uma acurácia não tão certa.

## 6. Referências Bibliográficas

- Notas de Aula da disciplina disponibilizados pelo professor por meio do *e-mail*.
- Aulas ministradas pelo professor em sala de aula e laboratório.
- <http://computacaointeligente.com.br/artigos/classificacao-de-dados>
- <https://archive.ics.uci.edu/ml/datasets/iris>
- [https://scikit-learn.org/stable/auto\\_examples/datasets/plot\\_digits\\_last\\_image.html](https://scikit-learn.org/stable/auto_examples/datasets/plot_digits_last_image.html)
- <https://archive.ics.uci.edu/ml/datasets/Pen-Based+Recognition+of+Handwritten+Digits>
- <https://archive.ics.uci.edu/ml/datasets/wine>
- [https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Diagnostic\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic))