Relatório referente ao segundo trabalho de implementação da matéria de Inteligência Artificial

Rafael Belmock Pedruzzi

Universidade Federal do Espírito Santo - Vitória

Abstract

Neste relatório realizaremos uma comparação experimental entre um conjunto pré-definido de técnicas de aprendizado e classificação automática aplicadas a alguns problemas de classificação.

1. Introdução

Recentemente a linguagem de programação Python tem ganhado cada vez mais popularidade para a implementação de aplicações baseadas em inteligência artificial. A principal razão para isso é a nova biblioteca de funções feitas para esse propósito chamada scikit-learn[1]. Esta biblioteca possui diversos mecanismos que suportam o desenvolvimento de programas inteligentes, como ferramentas de classificação, validação, predição, etc.

Neste relatório realizaremos uma comparação entre alguns métodos de classificação utilizando Python e o scikit-learn. Os classificadores avaliados serão ZeroR, OneR, OneR Probabilístico (Probabilistic OneR), Centroide (Centroid), Centroide OneR (Centroid OneR), Naive Bayes Gaussiano (Gaussian Naive Bayes - GNB), KNeighborsClassifier (Knn), Árvore de Decisão (Decision Tree), Rede Neural (Multi-Layer Perceptron - MLP) e Florestas de Árvores (Random Forest). Os últimos cinco são métodos já implementados pela biblioteca e suas referências podem ser encontradas, respectivamente, através dos seguintes links:

- `https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html#sklearn.naive_bayes.GaussianNB`

- https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
- https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier
- https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#sklearn.neural_network.MLPClassifier
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn.ensemble.RandomForestClassifier

Os demais métodos foram implementados e serão propriamente descritos na seção 3.

Esses métodos serão testados em quatro bases de dados (datasets) diferentes proporcionadas pelo scikit-learn para fins de teste. Serão elas iris, digits, wine e breast cancer. Para cada uma será executado o algorítimo de comparação, descrito na seção 4, para cada métodos. Os próprios bancos serão descritos na seção 2

Para realização dos experimentos, estes métodos foram divididos em dois grupos distintos: Os métodos que não necessitam de ajuste de hiperparâmetros, a saber ZeroR, OneR, OneR Probabilístico, Centroide, Centroide OneR e GNB, e os que necessitam (Knn, Árvore de Decisão, Rede Neural e Florestas de Árvores). O algorítimo de comparação também será dividido em duas etapas, uma para cada grupo.

2. Os Datasets

As bases de dados fornecidas pelo scikit-learn são conjuntos de amostras prefabricadas utilizadas para testes. Cada conjunto possui um determinado número de exemplos já classificados que podem ser divididos em conjuntos para teste e validação de classificadores. Quatro bases serão utilizadas, sendo elas iris, que possui 3 classes de 50 amostras cada, digits, composta de 10 classes com cerca de 180 amostras cada, wine, com 3 classes de 59, 71 e 48 amostras totalizando 178 exemplos e breast cancer, com duas classes de 212 e 357 amostras totalizando 569 exemplos.

3. Os Classificadores Implementados

Nessa seção descreveremos os classificadores que necessitaram ser implementados. Todos seguem um modelo compatível com as operações do scikit-learn. A especificação do modelo pode ser encontrada através do link `https://scikit-learn.org/dev/developers/develop.html`. É válido destacar que nenhum desses classificadores necessita de hiperparâmetros, portanto, não participando da etapa de busca em grade.

3.1. ZeroR

O mais simples desenvolvido, ZeroR é um classificador que não utiliza nenhuma propriedade interna dos objetos avaliados. Ele simplesmente classifica todos os casos de entrada na classe majoritária do conjunto de treino. O ZeroR apresenta péssimo desempenho e normalmente é utilizado apenas como medida comparativa, apesar de também ser pouco eficiente para isso como veremos a seguir.

3.2. OneR

Utilizando apenas uma única propriedade, o método OneR consiste em encontrar a propriedade que melhor represente as diversas classes e inferi-las a partir dessa propriedade. Como as bases de dados utilizadas possuem propriedades continuas optou-se por discretiza-las segundo o número de elementos em uma quantidade de conjuntos igual ao número de classes.

Apesar de ser superior ao ZeroR, este método também é considerado pouco eficiente e usado normalmente como medida comparativa.

3.3. OneR Probabilístico

Similar ao método OneR com apenas uma diferença significativa: A propriedade selecionada não define exatamente a classificação mas sim uma probabilidade de ser classificado nessa classe. Nessa abordagem todas as possíveis classes identificadas ao analisar a propriedade escolhida recebem uma chance de

serem selecionadas baseada na frequência de ocorrência daquele valor da propriedade nessa classe. Assim, a classe mais provável recebe maior chance de ser selecionada, sem descartar a possibilidade de escolher as demais classes.

3.4. Centroide

No método Centroide cada objeto é considerado como um ponto cartesiano de n dimensões. Ao executar o método *fit* a entrada é dividida segundo as classes e então é calculado o centroide de cada classe. Para executar a classificação, calcula-se a distância cartesiana do objeto para cada centroide. Ele será, então, classificado na classe cujo centroide for o mais próximo.

3.5. Centroide OneR

O método Centroide OneR é análogo ao método Centroide com a diferença de que, ao dividir a entrada, utiliza-se conjuntos formados a partir das classes reais e de classes estimadas pelo método OneR ao invés de somente as classes reais.

4. Os Experimentos Realizados

Nesta seção descreveremos os experimentos realizados, apresentando o algorítimo de comparação e os resultados alcançados.

4.1. Apresentação dos Algorítimos

Como explicado anteriormente, devido as necessidades dos classificadores testados, a execução foi dividida em duas etapas. A primeira etapa consiste no treino e teste com validação cruzada de 10 *folds* dos classificadores que não possuem hiperparâmetros, isto é, os classificadores ZeroR, OneR, OneR Probabilístico, Centroide, Centroide OneR e Naive Bayes Gaussiano. E a segunda etapa, no treino, validação e teste dos classificadores que precisam de ajuste de hiperparâmetros, isto é, os classificadores Knn, Árvore de Decisão, Redes Neurais e Florestas de Árvores. Neste caso o procedimento de treinamento, validação e teste será realizado através de ciclos aninhados de validação e teste, com o ciclo interno de validação contendo 4 *folds* e o externo de teste com 10 *folds*.

O seguinte pseudo-código descreve o algorítimo utilizado na primeira etapa:

Algorithm 1 Fase 1

```
para cada Dataset faça
    para cada classificador que não precisa de ajuste de hiperparâmetros faça
        executar treino e teste com validação cruzada de 10 folds
        armazenar média e desvio padrão das acurácias de cada fold
    fim para
    gerar tabela contendo as média e desvio padrão de cada classificador
    gerar bloxpot dos resultados de cada classificador
fim para
```

E para a segunda etapa:

Algorithm 2 Fase 2

para cada Dataset **faça**
    para cada classificador que precisa de ajuste de hiperparâmetros **faça**
        executar busca em grade de validação com 4 *folds*
        executar teste com validação cruzada de 10 *folds*
        armazenar média e desvio padrão das acurácias de cada *fold* do ciclo de teste
    fim para
    gerar tabela contendo as média e desvio padrão de cada classificador
    gerar bloxpot dos resultados de cada classificador
fim para

4.2. Apresentação dos Resultados

A seguir serão apresentados as tabelas contendo as médias e desvio padrão das acurácias de cada *fold* de teste e os bloxpots das acurácias alcançadas em cada *fold* de teste de cada etapa gerados pelo algorítimo apresentado para cada base de dados.

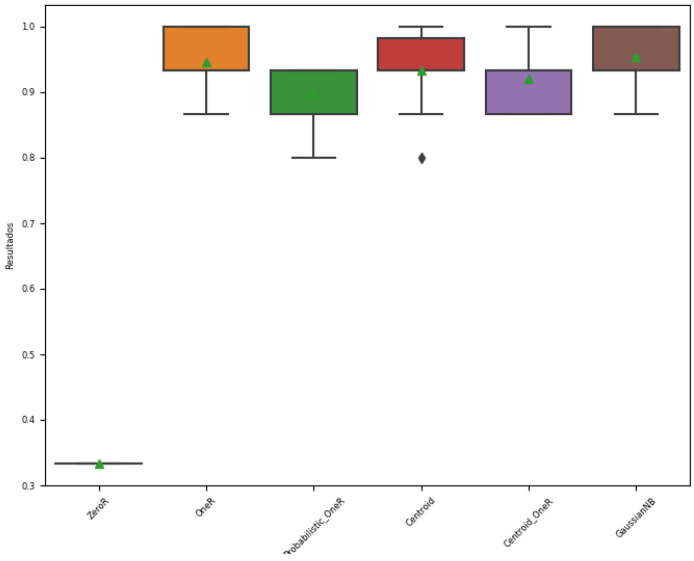
4.2.1. Iris

Primeira Etapa:

Tabela 1: Fase 1 - Iris

Classificador	Mdia das Acurcias	DP das Acurcias
ZeroR	0.333333	5.55112e-17
OneR	0.946667	0.0498888
Probabilistic_OneR	0.9	0.0447214
Centroid	0.933333	0.0596285
Centroid_OneR	0.92	0.0498888
GaussianNB	0.953333	0.0426875

Figura 1: Fase 1 - iris

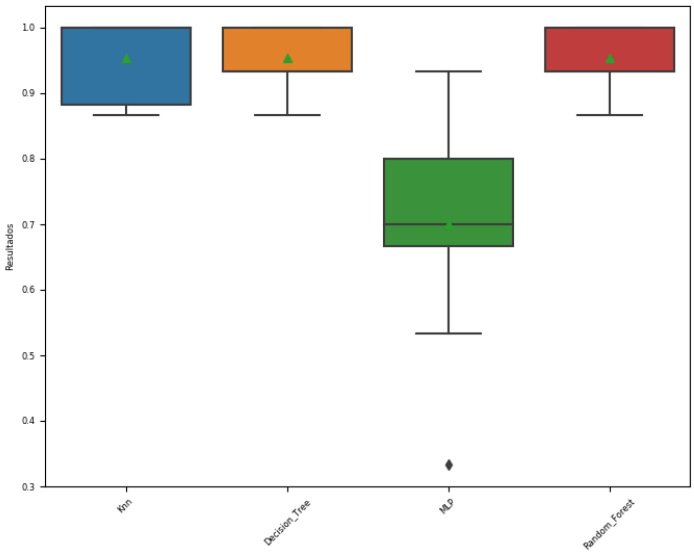


Segunda Etapa:

Tabela 2: Fase 2 - Iris

Classificador	Média das Acurácias	DP das Acurácias
Knn	0.953333	0.06
Decision_Tree	0.953333	0.0426875
MLP	0.7	0.163978
Random_Forest	0.953333	0.0426875

Figura 2: Fase 2 - iris



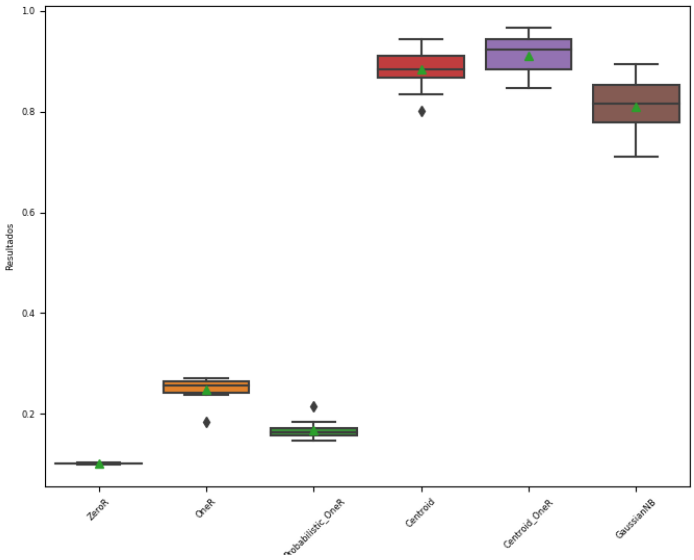
4.2.2. Digits

Primeira Etapa:

Tabela 3: Fase 1 - Digits

Classificador	Mdia das Acurcias	DP das Acurcias
ZeroR	0.101274	0.0012744
OneR	0.248686	0.024544
Probabilistic_OneR	0.168161	0.0184815
Centroid	0.88361	0.0411268
Centroid_OneR	0.911596	0.0403568
GaussianNB	0.810354	0.0566554

Figura 3: Fase 1 - Digits

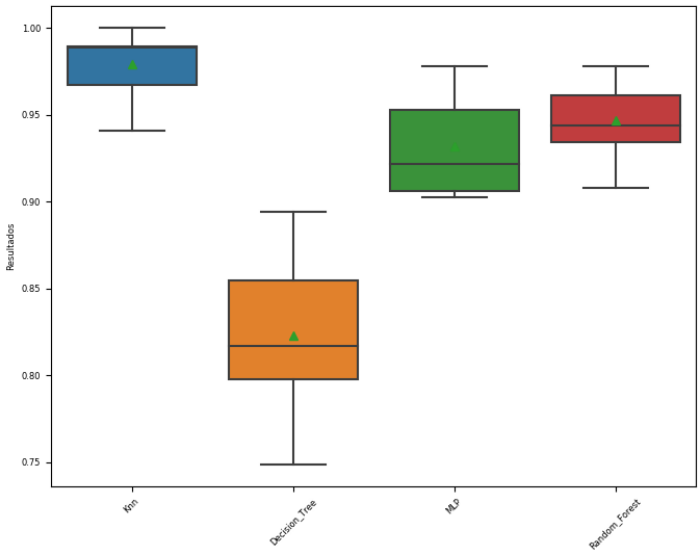


Segunda Etapa:

Tabela 4: Fase 2 - Digits

Classificador	Média das Acurácias	DP das Acurácias
Knn	0.978894	0.0176012
Decision_Tree	0.823026	0.0424116
MLP	0.931486	0.0273966
Random_Forest	0.946659	0.0203771

Figura 4: Fase 2 - Digits



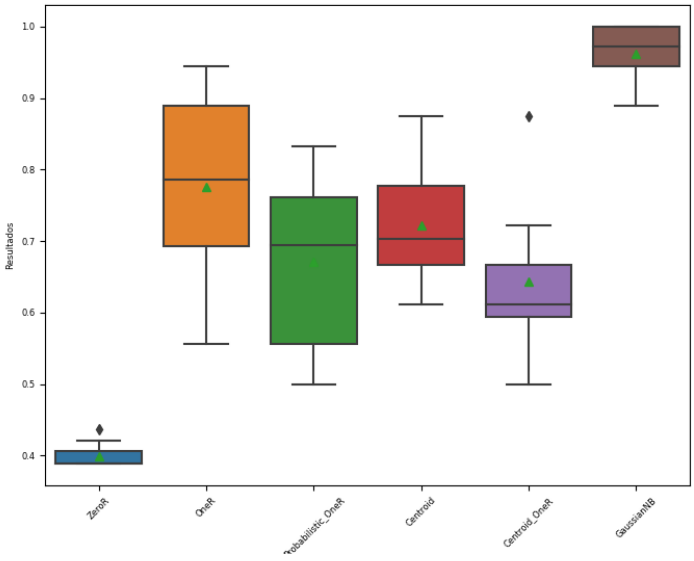
4.2.3. Wine

Primeira Etapa:

Tabela 5: Fase 1 - Wine

Classificador	Mdia das Acurcias	DP das Acurcias
ZeroR	0.399254	0.0168716
OneR	0.775774	0.125156
Probabilistic_OneR	0.670769	0.121325
Centroid	0.721607	0.0849013
Centroid_OneR	0.643107	0.0958237
GaussianNB	0.961696	0.042442

Figura 5: Fase 1 - Wine

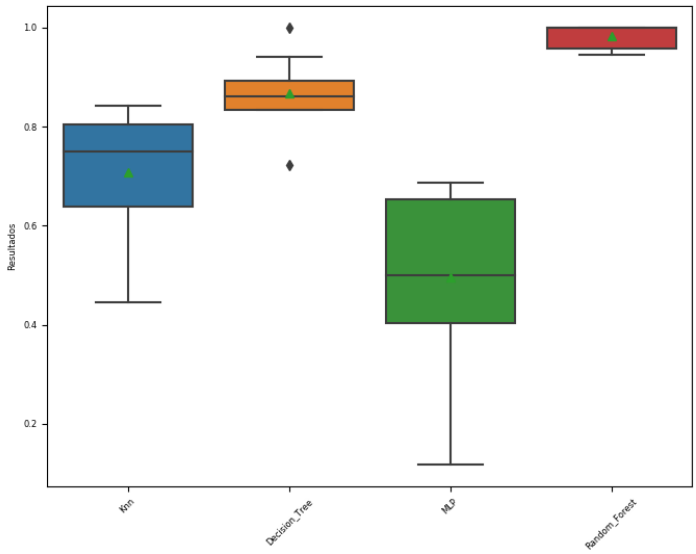


Segunda Etapa:

Tabela 6: Fase 2 - Wine

Classificador	Média das Acurácias	DP das Acurácias
Knn	0.708925	0.124287
Decision_Tree	0.866925	0.0710022
MLP	0.495135	0.167857
Random_Forest	0.983333	0.0254588

Figura 6: Fase 2 - Wine



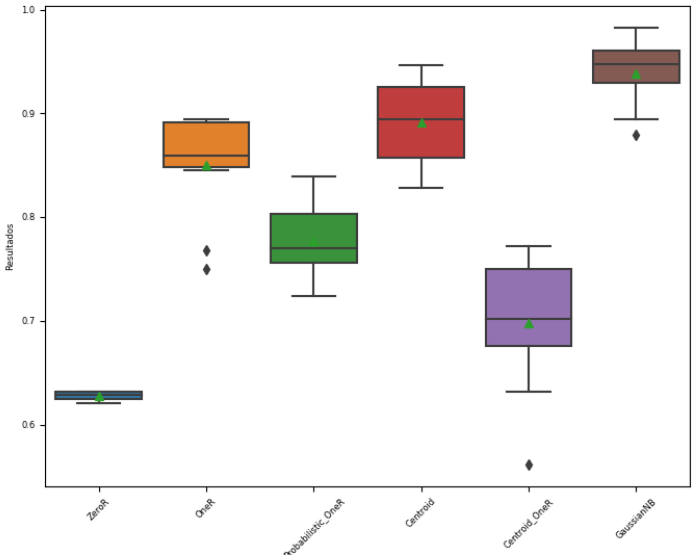
4.2.4. Breast Cancer

Primeira Etapa:

Tabela 7: Fase 1 - Breast Cancer

Classificador	Mdia das Acurcias	DP das Acurcias
ZeroR	0.627427	0.00441189
OneR	0.850265	0.048861
Probabilistic_OneR	0.776941	0.0360045
Centroid	0.891364	0.0387938
Centroid_OneR	0.697986	0.0618894
GaussianNB	0.93868	0.0301129

Figura 7: Fase 1 - Breast Cancer

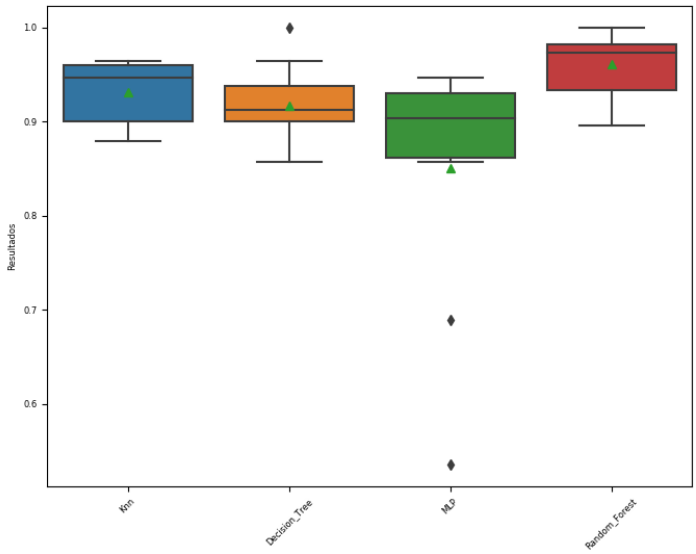


Segunda Etapa:

Tabela 8: Fase 2 - Breast Cancer

Classificador	Média das Acurácias	DP das Acurácias
Knn	0.931659	0.0312452
Decision_Tree	0.917532	0.0417313
MLP	0.85032	0.126403
Random_Forest	0.96149	0.0327803

Figura 8: Fase 1 - Breast Cancer



5. Conclusões

No decorrer deste relatório pudemos testar o desempenho de diversos classificadores e verificamos que houve bastante divergência quanto a eficiência mesmo para bases de dados relativamente simples. Algumas informações notórias podem ser identificadas analisando os resultados. Primeiramente, como já era esperado, o classificador ZeroR apresentou o pior resultado em todos os casos. Também podemos perceber que o método OneR Probabilístico foi sempre inferior ao OneR. Isso pode ser explicado pelo fato de que, apesar do OneR Probabilístico possuir uma classificação mais flexível que permite a correção de elementos destoantes, as chances de que esses elementos sejam corretamente classificados é muito baixa. Além disso ele também permite a classificação errônea de um elemento pertencente ao caso mais geral. Isto pode indicar que algorítimos probabilísticos podem não ser adequados para problemas de classificação pois podem introduzir mais erros.

Obviamente aproveitar o máximo possível de informações da entrada é a melhor abordagem para classificar elementos e os resultados comprovam isto com uma visível superioridade do classificador Centroide sobre os ZeroR, OneR e OneR Probabilístico. Ele também foi mais eficiente que o Centroide OneR para a maioria dos casos. Entretanto os melhores resultados foram os encontrados pelos métodos nativos do skikit-learn. Particularmente o GNB e a Floresta Aleatória foram os únicos métodos com mais de 90% de acurácia para todos os casos enquanto os demais apresentaram variações.

Com isto podemos concluir que para os casos testados os classificadores GNB e Floresta Aleatória foram os mais eficientes e, portanto, são os mais confiáveis para aplicações diversas. Entretanto é possível que hajam contextos não testados que contradigam esta afirmação. Possíveis trabalhos futuros podem incluir maior abrangência de bases de dados diversificadas.

Referências

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