# Resultados dos Algoritmo KNN e DT com Diferentes Conjuntos de Dados

**Felipe Archanjo da Cunha Mendes**[1]

[1]Bacharelado em Ciência da Computação – Universidade Tecnológica Federal do Paraná (UFRGS)
Campo Mourão – PR – Brasil

felipemendes.1999@alunos.utfpr.edu.br

***Abstract.*** *This report presents the results of experiments conducted using the k-Nearest Neighbors (k-NN) and Decision Trees algorithms on three different datasets. The k-NN algorithm was used with different values of k and Euclidean distance as the distance metric. The Decision Trees algorithm was used with and without pruning. The datasets used in the experiments were the digits dataset, the spam dataset, and the writer dataset. The results were evaluated using both cross-validation and a test set. The performance of each algorithm was measured in terms of accuracy.*

*The experiments showed that the k-NN algorithm performed well on all three datasets, with high accuracy rates achieved for various values of k. The Decision Trees algorithm, on the other hand, did not perform as well, with lower accuracy rates observed for all three datasets. Pruning did not improve the performance of the algorithm significantly.*

***Resumo.*** *Este relatório apresenta os resultados de experimentos realizados usando os algoritmos k-Nearest Neighbors (k-NN) e Decision Trees em três conjuntos de dados diferentes. O algoritmo k-NN foi usado com diferentes valores de k e a distância Euclidiana como métrica de distância. O algoritmo Decision Trees foi usado com e sem poda. Os conjuntos de dados usados nos experimentos foram o conjunto de dígitos, o conjunto de spam e o conjunto de escritores. Os resultados foram avaliados usando validação cruzada e um conjunto de teste. O desempenho de cada algoritmo foi medido em termos de precisão.*

*Os experimentos mostraram que o algoritmo k-NN teve um bom desempenho em todos os três conjuntos de dados, com altas taxas de precisão alcançadas para vários valores de k. O algoritmo Decision Trees, por outro lado, não teve um desempenho tão bom, com taxas de precisão mais baixas observadas para os três conjuntos de dados. A poda não melhorou significativamente o desempenho do algoritmo.*

## 1. Introdução

A mineração de dados e o aprendizado de máquina são técnicas importantes para a extração de informações úteis de grandes conjuntos de dados. Uma das abordagens mais comuns de aprendizado de máquina é o algoritmo k-NN (k-Nearest Neighbors), que é um algoritmo baseado em instância e não paramétrico utilizado para classificação e regressão. Outra abordagem comum é a utilização de árvores de decisão, que são

modelos de classificação que utilizam uma série de regras de decisão para a classificação de instâncias.

Neste artigo, serão descritos os experimentos realizados com os algoritmos k-NN e árvore de decisão utilizando a ferramenta Weka. Serão utilizados diferentes conjuntos de dados, incluindo conjuntos de dados para reconhecimento de dígitos manuscritos, classificação de e-mails como spam ou não spam e reconhecimento de caracteres manuscritos. Para o algoritmo k-NN, serão avaliados diferentes valores de k e distâncias euclidianas, enquanto para a árvore de decisão, será avaliado o desempenho com e sem poda. Serão apresentados os parâmetros utilizados nos experimentos e as taxas de acerto alcançadas em cada conjunto de dados.

## 2. Se Familiarizando aos dados

Os arquivos mencionados são conjuntos de dados em formato ARFF, um formato comum para armazenar conjuntos de dados utilizados em mineração de dados e aprendizado de máquina.

O arquivo "digTreino.arff" representa um conjunto de dados para reconhecimento de dígitos manuscritos, com 65 características, 10 classes e um total de 382 amostras por classe. O arquivo "digTeste.arff" também representa um conjunto de dados para reconhecimento de dígitos manuscritos, com as mesmas 65 características, 10 classes e um total de 182 amostras por classe.

O arquivo "spambase-train.arff" representa um conjunto de dados para classificação de e-mails como spam ou não spam, com 58 características, 2 classes e um total de 509 amostras na classe negativa e 871 amostras na classe positiva. O arquivo "spambase-test.arff" representa um conjunto de dados de teste para o mesmo problema, com as mesmas 58 características, 2 classes e um total de 402 amostras na classe negativa e 702 amostras na classe positiva.

O arquivo "writerTrain.arff" representa um conjunto de dados para reconhecimento de caracteres manuscritos, com 60 características, 250 classes e um número variável de amostras por classe (entre 0 e 2). O arquivo "writerTest.arff" representa um conjunto de dados de teste para o mesmo problema, com as mesmas 60 características, 250 classes e um número variável de amostras por classe (entre 0 e 1).

## 3. KNN

Para o conjunto de dados digitos, os resultados obtidos utilizando validação cruzada indicam que o melhor desempenho foi alcançado com k=3, com uma taxa de acerto de 98,6137%. No entanto, ao utilizar o arquivo de teste, o melhor resultado foi obtido com k=1, com uma taxa de acerto de 97,941%.

No conjunto de dados spambase, os resultados mostram que a taxa de acerto é alta quando k=1, tanto na validação cruzada (93,8406%) quanto no arquivo de teste (100%).

No entanto, quando se utiliza um valor maior para k, a taxa de acerto diminui consideravelmente, chegando a 88,7681% na validação cruzada com k=5 e 93,1159% no arquivo de teste com k=7.

Para o conjunto de dados writer, os resultados indicam que o algoritmo k-NN tem um desempenho muito ruim. A taxa de acerto na validação cruzada varia de 91,2821% para k=1 a apenas 0,5128% para k=7. No arquivo de teste, o melhor resultado obtido foi de apenas 93,8462% para k=1.

Em geral, pode-se observar que o desempenho do algoritmo k-NN varia significativamente de acordo com o conjunto de dados e com o valor de k escolhido. Para o conjunto digitos, k=3 apresentou o melhor desempenho na validação cruzada, enquanto k=1 foi melhor no arquivo de teste. Para o conjunto spambase, k=1 foi o melhor em ambos os casos. Para o conjunto writer, o algoritmo k-NN apresentou um desempenho muito ruim em geral.

## 4. DT

Para o conjunto de dados "digitos", o algoritmo Decision Trees obteve uma taxa de acerto de 89,72% utilizando a validação cruzada com a árvore não podada e 85,75% no conjunto de teste. Quando a poda foi aplicada, a taxa de acerto na validação cruzada permaneceu praticamente inalterada em 89,69%, enquanto a taxa de acerto no conjunto de teste apresentou uma leve redução para 85,69%.

Para o conjunto de dados "spam", o algoritmo Decision Trees obteve uma taxa de acerto de 93,26% utilizando a validação cruzada com a árvore não podada e 97,46% no conjunto de teste. Quando a poda foi aplicada, a taxa de acerto na validação cruzada aumentou para 93,84%, enquanto a taxa de acerto no conjunto de teste permaneceu praticamente inalterada em 98,01%.

Para o conjunto de dados "writer", o algoritmo Decision Trees obteve uma taxa de acerto de 10,51% utilizando a validação cruzada com a árvore não podada e 55,89% no conjunto de teste. Quando a poda foi aplicada, a taxa de acerto na validação cruzada permaneceu inalterada em 10,51%, enquanto a taxa de acerto no conjunto de teste permaneceu a mesma. É importante ressaltar que o conjunto de dados "writer" apresentou um número muito baixo de amostras por classe, o que pode ter afetado a capacidade do algoritmo em aprender padrões relevantes para a classificação.

## 5. Conclusão

Neste relatório, foram apresentados os resultados de dois algoritmos de aprendizado de máquina aplicados a três conjuntos de dados diferentes. O algoritmo k-NN foi aplicado aos conjuntos de dados de dígitos escritos à mão, spambase e writer, enquanto o

algoritmo Decision Tree foi aplicado somente aos conjuntos de dados de dígitos e writer.

No caso do algoritmo k-NN, foi utilizado o método de validação cruzada com 10 folds e um arquivo de teste fornecido para avaliar o desempenho do modelo em diferentes valores de k. Os resultados mostraram que o valor de k = 3 obteve as melhores taxas de acerto para o conjunto de dados de dígitos, enquanto o valor de k = 1 obteve as melhores taxas de acerto para os conjuntos de dados de spambase e writer.

Já no caso do algoritmo Decision Tree, foi utilizado o método de validação cruzada com 10 folds e um arquivo de teste fornecido para avaliar o desempenho do modelo tanto com a poda quanto sem a poda. Os resultados mostraram que a aplicação da poda não melhorou significativamente as taxas de acerto para nenhum dos conjuntos de dados testados.

Em geral, pode-se concluir que os algoritmos de aprendizado de máquina apresentaram desempenhos satisfatórios nos conjuntos de dados testados, embora a seleção dos melhores parâmetros tenha variado para cada conjunto de dados. Além disso, é importante ressaltar que os resultados podem ser influenciados por fatores como a qualidade dos dados e a escolha das métricas de avaliação do modelo. Portanto, é importante considerar esses fatores ao selecionar e ajustar modelos de aprendizado de máquina para problemas reais.

## 6. Gráficos e Tabelas

| K | digitos - CV | digitos - Teste | spam - CV | spam - Teste | writer - CV | writer - Teste |
|---|---|---|---|---|---|---|
| **1** | 98,4567 | 97,941 | 93,8406 | 100 | 91,2821 | 93,8462 |
| **3** | 98,6137 | 97,8297 | 90,2174 | 95,471 | 18,7179 | 93,5897 |
| **5** | 98,5613 | 97,941 | 88,7681 | 93,1159 | 2,8205 | 53,8462 |
| **7** | 98,509 | 97,5515 | 88,1159 | 92,029 | 0,5128 | 39,4872 |
| **9** | 98,5352 | 97,8854 | 88,6232 | 90,6703 | 0 | 29,2308 |

**Tabela 1. Representa a influência que o tipo de treinamento dos dados e o valor de K tem com a acurácia final.**

| UNPRUNED | digitos - CV | digitos - Teste | spam - CV | spam - Teste | writer - CV | writer - Teste |
|---|---|---|---|---|---|---|

| | | | | | | |
|---|---|---|---|---|---|---|
| **FALSE** | 89,7201 | 85,754 | 93,2609 | 97,4638 | 10,5128 | 55,8974 |
| **TRUE** | 89,694 | 85,6984 | 93,8406 | 98,0072 | 10,5128 | 55,8974 |

**Tabela 2. Representa a influência que o tipo de treinamento dos dados e a poda tem com a acurácia final.**