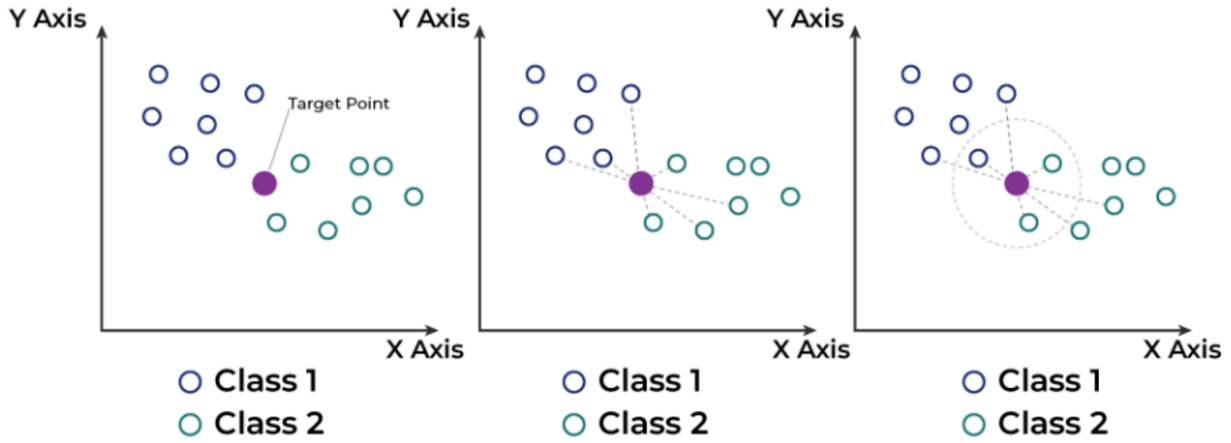
Algoritmo K-Nearest Neighbors

Comparação de Implementações: Hardcode vs Scikit-learn



GCC128 - Inteligência Artificial

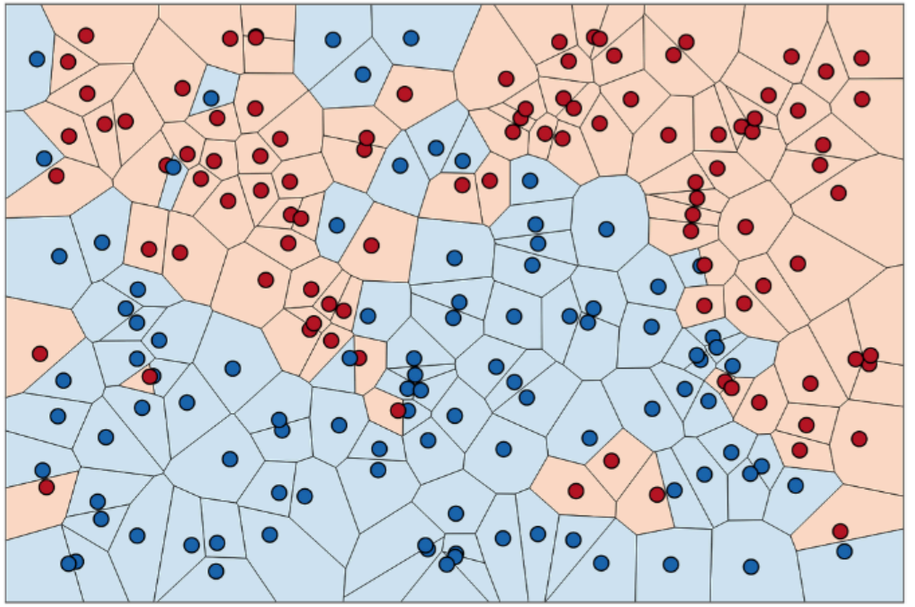
Sistemas de Informação – 14A

Gustavo de Jesus Teodoro - 202311146
Thiago Lima Pereira - 202310057

Lavras - MG

Introdução ao Algoritmo KNN

- **K-Nearest Neighbors (KNN)** é um algoritmo de classificação supervisionada baseado no princípio de proximidade entre pontos de dados.
- Funciona identificando os K vizinhos mais próximos de uma amostra e classificando-a conforme a classe predominante nesse grupo.
- Aplicável em problemas de classificação e regressão, com uso frequente em conjuntos de dados como o Iris.
- **Objetivo do trabalho:** comparar duas abordagens de implementação: A primeira abordagem se dá pela implementação "hardcore" do algoritmo, já a segunda se dá por meio de uma biblioteca chamada Scikit-learn, a qual oferece o algoritmo pronto

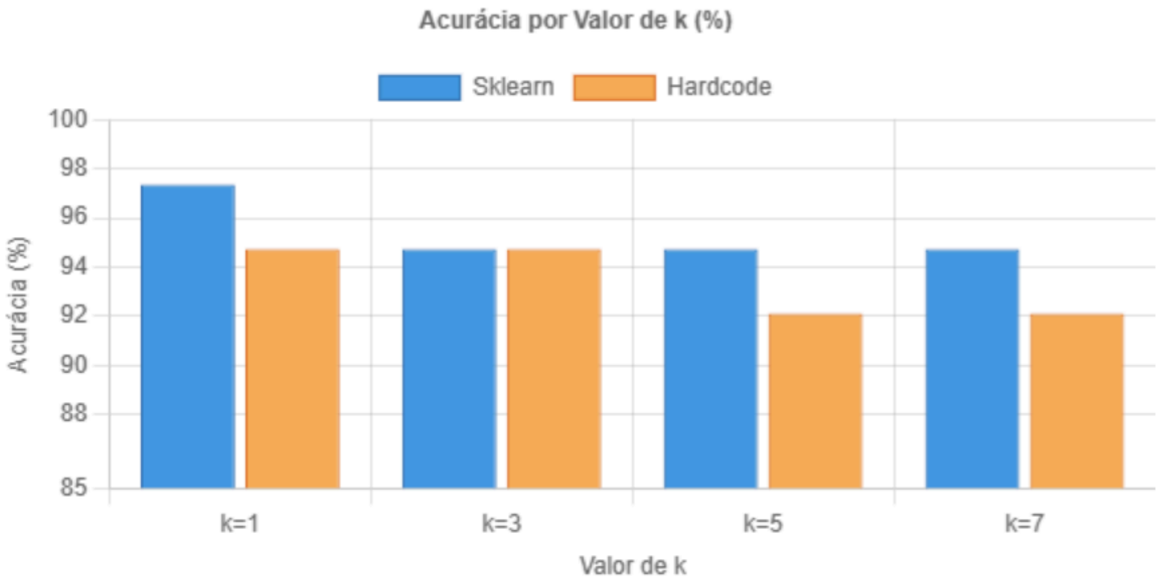


Metodologia de Comparação

- Métricas avaliadas para fins comparativos:
  - **Acurácia** - mede quantos elementos foram previstos de maneira correta durante os testes
  - **Precisão** - fornce informações relacionadas a quantidade de acertos obtidas em cada uma das classes
  - **Revocação** - tem como intuito medir quais foram as previsões incorretas em cada uma das classes
- Dataset: Iris (150 amostras, 3 classes, 4 features)
- Valores de k testados: 1, 3, 5 e 7

Resultados de Acurácia

- **Sklearn:** atingiu melhor acurácia para k=1 (97,37%)
- **Hardcode:** chegou a 94,74% para k=1, com queda para k=5 e k=7 (92,11%)
- **Empate em k=3:** ambas implementações com 94,74%
- **Desempenho mais estável:** scikit-learn manteve acurácia de 94,74% para k=3, k=5 e k=7
- Implementação Hardcode sofreu queda de desempenho com o aumento de k, enquanto a versão Sklearn manteve consistência



Comparação de acurácia entre implementações para diferentes valores de k

Precisão por Classe

- Análise detalhada da precisão: desempenho por classe e implementação
- Iris-setosa: Ambas as implementações alcançaram 100% de precisão para todos os valores de k, demonstrando ser uma classe facilmente separável.
- Iris-versicolor: Hardcode entre 90% e 90.91%, enquanto Sklearn manteve estabilidade entre 92% e 93%.
- Iris-virginica: Maior diferença observada. Sklearn manteve consistência com 100% (k=1) e 92% (demais), enquanto Hardcode caiu até 83.33% para k=5 e k=7.

Comparativo de Precisão por Classe e Valor de k

Classe	Implementação	k=1	k=3	k=5	k=7
Iris-setosa	Hardcode	100%	100%	100%	100%
	Sklearn	100%	100%	100%	100%
Iris-versicolor	Hardcode	90.91%	90.91%	90%	90%
	Sklearn	93%	92%	92%	92%
Iris-virginica	Hardcode	90.91%	90.91%	83.33%	83.33%
	Sklearn	100%	92%	92%	92%

Revocação por Classe

- Iris-setosa: Ambas implementações alcançaram 100% de revocação para todos os valores de k, mostrando forte separabilidade desta classe.
- Iris-versicolor: A versão Hardcode apresentou queda significativa de 90,91% (k=1,3) para 81,82% (k=5,7).
- Já o Sklearn manteve desempenho mais estável, com revocação de 100% para k=1 e 92% para os demais valores de k.
- Iris-virginica: A revocação da versão Hardcode foi constante em 90,91% para todos os valores de k. O Sklearn apresentou revocação de 92% em todos os casos.
- A análise de revocação confirma a maior consistência da implementação do Sklearn, especialmente para classes com maior sobreposição de características.

Classe	Implementação	k=1	k=3	k=5	k=7
Iris-setosa	Hardcode	100%	100%	100%	100%
	Sklearn	100%	100%	100%	100%
Iris-versicolor	Hardcode	90,91%	90,91%	81,82%	81,82%
	Sklearn	100%	92%	92%	92%
Iris-virginica	Hardcode	90,91%	90,91%	90,91%	90,91%
	Sklearn	92%	92%	92%	92%

Comparativo Geral das Implementações

- **Sklearn:** Resultados mais robustos e estáveis em diferentes valores de k.
- **Hardcode:** Maior variação de desempenho, especialmente com aumento de k.
- **Recomendação:** Para produção, utilizar bibliotecas otimizadas como scikit-learn.

Característica	Hardcode	Scikit-learn
Valor didático	✓ Alto	✓ Médio
Acurácia (k=1)	94,74%	97,37%
Estabilidade	⚠ Menor	✓ Maior
Facilidade de uso	⚠ Baixa	✓ Alta
Recomendado para	Aprendizado	Produção

Conclusão e Referências

Principais Conclusões

- **Robustez do scikit-learn:** Resultados mais estáveis e consistentes em diferentes valores de k.
- **Implementação Hardcode:** Valiosa para fins didáticos e compreensão interna do algoritmo.
- **Métricas:** scikit-learn apresentou melhor acurácia, precisão e revocação na maioria dos casos.
- **Recomendação:** Para aplicações práticas, recomenda-se o uso de bibliotecas otimizadas como scikit-learn.

> *"Para fins de aprendizado, a implementação Hardcode é um exercício fundamental. Para aplicações práticas, o uso de bibliotecas como scikit-learn se demonstrou mais recomendado."*

Referências

Dataset Iris

Fisher, R.A. "The use of multiple measurements in taxonomic problems"

Documentação scikit-learn

https://scikit-learn.org/stable/scikit-learn.org - Módulo KNeighborsClassifier

Materiais da disciplina

GCC128 - Inteligência Artificial, UFLA

Link do vídeo

https://drive.google.com/file/d/1uXH8Ssohl_5b88Il7niYqnXUTUrOKIyR/view?usp=sharing