UNIVERSIDADE FEDERAL DE LAVRAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Aluno: Michel Alexandrino de Souza **Matrícula: 202111084**
Aluno: João Pedro Alves Carneiro Valadão **Matrícula: 202020295**

GCC128 – Inteligência Artificial **Prof. Ahmed Ali Abdalla Esmin**

Relatório Técnico – KNN (K-Nearest-Neighbors)

1. Definição e apresentação da base de dados

O conjunto de dados Iris, introduzido por Ronald Fisher em 1936, constitui um benchmark clássico em machine learning, frequentemente utilizado para tarefas de classificação supervisionada. Composto por 150 amostras igualmente distribuídas entre três espécies do gênero Iris (setosa, versicolor e virginica), o dataset apresenta quatro atributos numéricos contínuos: comprimento e largura da sépala (em cm), e comprimento e largura da pétala (em cm). Cada instância está perfeitamente categorizada, sem valores faltantes ou inconsistentes, o que o torna ideal para prototipagem de algoritmos. A relevância científica desta base reside em suas propriedades estatísticas bem comportadas e na separabilidade parcial das classes. Enquanto Iris setosa é linearmente separável das demais, versicolor e virginica apresentam regiões de sobreposição em seu espaço de características, criando um desafio interessante para classificadores. A distribuição balanceada (50 amostras por classe) elimina a necessidade de técnicas de reamostragem, permitindo focar no desenvolvimento do modelo preditivo. A abordagem para classificação utilizada neste dataset foi o algoritmo KNN.

O algoritmo KNN (K-Nearest Neighbors) é uma técnica de aprendizado supervisionado baseada em instâncias, amplamente empregada em problemas de classificação e regressão. Sua principal característica é a ausência de uma fase explícita de treinamento: em vez de construir um modelo estatístico ou paramétrico, o KNN armazena todo o conjunto de treinamento e realiza as predições com base na similaridade entre as instâncias. Para classificar uma nova amostra, o KNN identifica os k exemplos mais próximos no espaço de características — os “vizinhos mais próximos” — e determina a classe mais frequente entre eles. A escolha do valor de k é crucial: valores baixos tornam o modelo sensível a ruídos, enquanto valores altos podem suavizar excessivamente as fronteiras de decisão, afetando a

acurácia. A medida de proximidade mais comum, e utilizada neste trabalho, é a **distância euclidiana**, definida matematicamente como:

$$d\,(p,q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

onde p e q são vetores n-dimensionais representando duas amostras, e p_i, q_i são os valores dos atributos na i-ésima dimensão. No caso do conjunto Iris, essa métrica é aplicada sobre os quatro atributos contínuos, permitindo estimar o grau de similaridade entre flores distintas com base em suas medidas morfológicas. Essa abordagem não assume hipóteses sobre a distribuição dos dados, o que, aliado à simplicidade da implementação, torna o KNN uma escolha natural para testes iniciais com o dataset Iris. Contudo, sua eficácia depende diretamente da normalização dos atributos — dado que a distância euclidiana é sensível à escala das variáveis — e da densidade amostral local em torno de cada instância.

2. Comparação entre as aplicações

Neste trabalho, foram implementadas duas abordagens distintas para o algoritmo KNN: uma versão manual (hardcore) e uma versão baseada na biblioteca scikit-learn. Ambas as implementações foram avaliadas utilizando o conjunto de dados Iris, com o objetivo de comparar desempenho, precisão e limitações de cada abordagem.

Implementação Hardcore

A versão manual do KNN foi desenvolvida do zero, utilizando Python e bibliotecas como numpy e pandas. Nesta abordagem, a distância euclidiana entre as amostras foi calculada explicitamente, e os vizinhos mais próximos foram identificados por meio de ordenação. A classificação foi realizada determinando a classe mais frequente entre os k-vizinhos mais próximos. Embora esta implementação ofereça total controle sobre o funcionamento do algoritmo, ela apresenta limitações em termos de desempenho, especialmente para conjuntos de dados maiores, devido à ausência de otimizações como estruturas de dados especializadas (ex.: árvores KD).

Implementação com Scikit-learn

A versão baseada no scikit-learn utilizou a classe KNeighborsClassifier, que é altamente otimizada e implementada em Cython. Esta abordagem oferece suporte a estruturas de dados eficientes, como árvores KD e Ball Trees, para acelerar a busca por vizinhos mais

próximos. Além disso, a biblioteca abstrai detalhes de implementação, permitindo maior foco na análise dos resultados. No entanto, essa simplicidade vem ao custo de menor flexibilidade para personalizações específicas.

Comparação de Desempenho

Os tempos de execução foram medidos para ambas as implementações, considerando os valores de k = [1, 3, 5, 7]. A implementação com scikit-learn apresentou tempos significativamente menores, devido às otimizações internas da biblioteca. Por outro lado, a versão hardcore, embora mais lenta, permitiu maior transparência no processo de classificação, sendo útil para fins educacionais e de validação.

Comparação de Métricas

Ambas as implementações produziram resultados consistentes em termos de acurácia, precisão e revocação, indicando que a lógica do algoritmo foi corretamente replicada na versão manual. No entanto, a implementação hardcore é mais suscetível a erros em datasets maiores ou mais complexos, devido à ausência de validações robustas e otimizações.

3. Resultados e limitações

Resultados

Os experimentos realizados demonstraram que o algoritmo KNN é eficaz para a classificação do conjunto de dados Iris, alcançando altas taxas de acurácia para diferentes valores de k. As matrizes de confusão geradas para cada valor de k evidenciaram que a classe Iris-setosa é perfeitamente separável, enquanto as classes Iris-versicolor e Iris-virginica apresentam maior sobreposição, resultando em erros de classificação entre elas.

A implementação com scikit-learn destacou-se pelo desempenho superior, com tempos de execução significativamente menores, especialmente para valores maiores de k. A implementação hardcore, por sua vez, foi útil para compreender os detalhes internos do algoritmo, mas apresentou limitações de escalabilidade.

Limitações

1. Dependência do Valor de k: A escolha do valor de k impacta diretamente o desempenho do modelo. Valores baixos tornam o modelo sensível a ruídos, enquanto valores altos podem suavizar excessivamente as fronteiras de decisão.

2. Escalabilidade: A implementação hardcore não é adequada para datasets maiores, devido à sua complexidade computacional $O(n^2)$ para a busca de vizinhos mais próximos. Já a versão com scikit-learn é mais escalável, mas ainda pode enfrentar limitações para datasets muito grandes.
3. Sensibilidade à Escala: A distância euclidiana, utilizada como métrica de proximidade, é sensível à escala dos atributos. Embora o conjunto Iris não exija normalização explícita, datasets com variáveis em escalas diferentes podem exigir pré-processamento adicional.
4. Sobreposição de Classes: A sobreposição entre Iris-versicolor e Iris-virginica limita a acurácia máxima alcançável, independentemente do valor de k ou da implementação utilizada.

Considerações Finais

Embora o KNN seja um algoritmo simples e eficaz para o conjunto de dados Iris, sua aplicabilidade em problemas reais depende de fatores como tamanho do dataset, dimensionalidade e separabilidade das classes. A implementação com scikit-learn é recomendada para aplicações práticas, enquanto a versão hardcore é mais adequada para fins educacionais e experimentais.