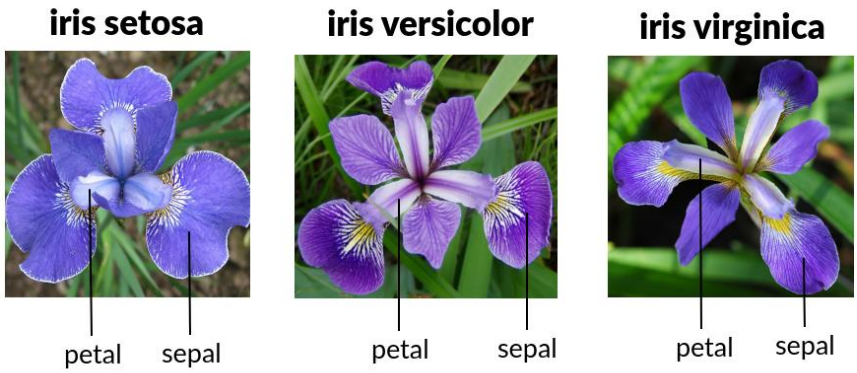
Classificador KNN para dataset de câncer de mama

Antônio Lima Della Flora

Introdução

A análise e classificação de dados são etapas cruciais em diversas áreas do conhecimento, e o aprendizado de máquina (Machine Learning) oferece ferramentas poderosas para essa finalidade. Entre os conjuntos de dados clássicos e amplamente utilizados para ilustrar e testar algoritmos de classificação, destaca-se o Iris Flower Dataset. Introduzido por Ronald Fisher em 1936, este dataset icônico serve como um benchmark fundamental para o desenvolvimento e avaliação de técnicas de reconhecimento de padrões.

O Iris Flower Dataset consiste em 150 amostras de três espécies diferentes de flores de íris: *Iris setosa*, *Iris versicolor* e *Iris virginica*. Para cada amostra, são medidas quatro características: o comprimento da sépala, a largura da sépala, o comprimento da pétala e a largura da pétala. O objetivo principal ao trabalhar com este dataset é construir modelos capazes de classificar corretamente cada amostra em sua respectiva espécie com base nessas medidas.



Metodologia

Pré processamento

O primeiro passo foi importar o dataset. Por ser um dataset bem famoso, ele já está incluso dentro da scikit-learn. Após carregar o dataset foi feito o pré processamento dos dados, dividindo os dados em 3 conjuntos:

- Conjunto de treino (60%): foi utilizado para treinar o modelo;
- Conjunto de validação (20%): Empregado para otimizar o hiper parâmetro 'k' do algoritmo KNN;
- Conjunto de testes (20%): Serviu para avaliar a performance final do modelo com o 'k' que entregou maior desempenho durante o treinamento. São dados que nunca foram utilizados para treinar o modelo.

A divisão dos conjuntos foi realizada utilizando o comando train_test_split da biblioteca sklearn. Para garantir uma divisão igual entre os 3 tipos de flores foi utilizada a estratificação com os tipos de flores do dataset. Também foi definido uma semente fixa para a divisão dos conjuntos, a fim de garantir a replicabilidade dos resultados.

O próximo passo foi normalizar os dados utilizando o *StandardScaler.* Ele foi calibrado somente com os dados do conjunto de treino para evitar vazamentos de informações dos conjuntos de validação e testes. Após calibrado todos os 3 conjuntos foram normalizados com as informações da média e desvio padrão do conjunto de treino.

A fim de testar o impacto da normalização, as próximas etapas foram realizadas com dois conjuntos, o normalizado e o conjunto inicial e suas performances serão exibidas lado a lado.

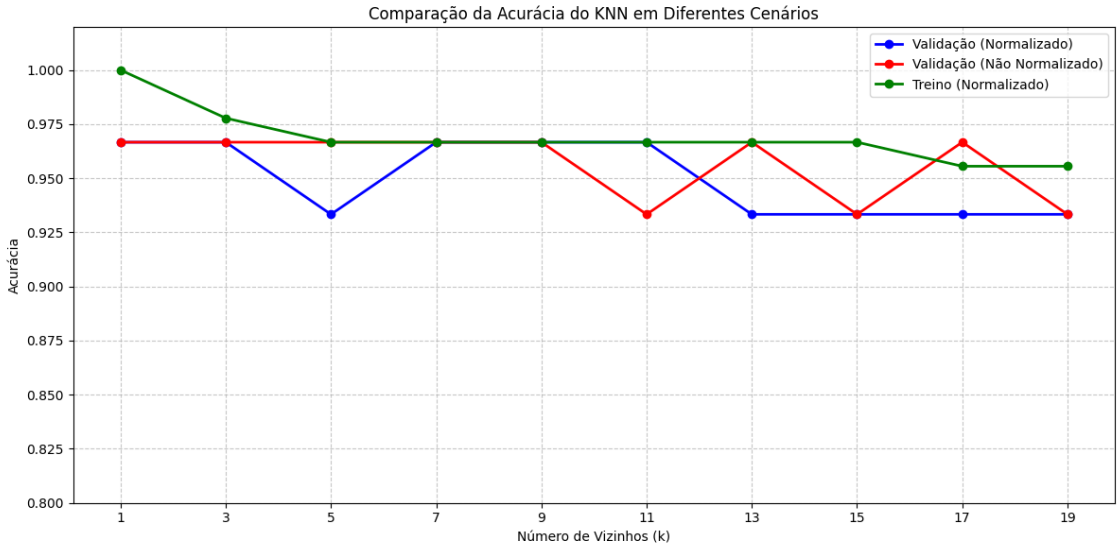
Otimização do hiper parâmetro

Com os dados já ajustados buscou-se descobrir qual é o valor ideal do hiper parâmetro 'k'. Para isso foram treinados modelos KNN com um k variável de 1 a 19 com incremento de 2 para evitar empates. O modelo foi treinado com o conjunto de treino e avaliado com o conjunto de validação.

Esse teste foi realizado tanto para os dados normalizados quando para os dados originais, para que seja possível comparar o impacto do pré-processamento no desempenho do modelo.

Também foi realizado o teste do conjunto de treino avaliando o próprio conjunto de treino. Esse teste serve para verificar se não há um overfitting no modelo.

A métrica de desempenho utilizada para classificar os modelos foi a acurácia, calculada com o accuracy_*score*. O resultado obtido está na figura abaixo.



Avaliação do Modelo Final

Com base nos resultados da etapa anterior, foi escolhido 9 como o melhor número de vizinhos para o modelo. Então foram criados dois modelos KNN, um utilizando os dados normalizados e outro não normalizados. Ambos com o 'k' igual a 9.

Esses modelos foram então submetidos ao conjunto de teste, que não foi utilizado em nenhum momento do treinamento.

Para avaliar o desempenho final do modelo foi utilizado o *classification_report,* que obtém métricas detalhadas de cada classe (cada tipo de flor) e a acurácia global.

As métricas obtidas para cada classe são as seguintes:

- Precisão (precision): é o número de verdadeiros positivos dividido pela quantidade de positivos detectados pelo modelo. É calculado com Verdadeiros Positivos / (Verdadeiros Positivos + Falsos Positivos). Se o valor foi baixo indica que o modelo possui muitos falsos positivos.
- Revocação (recall): Indica quantos % o modelo conseguiu identificar corretamente. É calculado com Verdadeiros Positivos / (Verdadeiros

Positivos + Falsos Negativos). Um baixo número indica que o modelo classificou muitas amostras como sendo de outras classes.

- Pontuação F1 (F1-Score): é a média harmônica da precisão e a revocação.
- Suporte (support): representa o número de ocorrências reais da classe. Ajuda a entender o volume de cada classe que foi utilizado para calcular as demais métricas.

Os resultados foram os seguintes:

Modelo com dados normalizados:

Classe	**precision**	**recall**	**1-score**	**support**
setosa	1.00	1.00	1.00	10
versicolor	0.909	1.00	0.952	10
virginica	1.00	0.90	0.947	10

Acurácia global: 0.9667

Modelo com dados não normalizados:

Classe	**precision**	**recall**	**1-score**	**support**
setosa	1.00	1.00	1.00	10
versicolor	0.909	1.00	0.952	10
virginica	1.00	0.90	0.947	10

Acurácia global: 0.9667

Conclusão

Os resultados mostram que ambos os modelos tiveram um bom desempenho. Ao analisar a tabela com as métricas, é possível perceber que o modelo classificou corretamente todas as 10 amostras da classe *setosa* e *versicolor*. Já a classe *virginica* teve um falso negativo em que a amostra foi classificada como *versicolor*.

A acurácia global de 0.967 para ambos os modelos reforça que o KNN conseguiu classificar de forma eficaz os diferentes tipos de flores.

É interessante notar que para o 'k' escolhido não houve diferença entre os modelos normalizados e não normalizados. Ambos os cenários tiveram resultados idênticos nas métricas. Isso pode sugerir que a escala das medidas do dataset não precisaram de muitas mudanças para ficarem normalizadas.

Por fim, buscou-se alguns artigos que utilizaram a mesma base de dados e treinaram um KNN para comparar a acurácia. Conforme a tabela abaixo, é possível

observar que o presente trabalho atingiu a mesma taxa de acerto do Asmita et al (2020) e performou melhor do que o Shivam (2019):

	Asmita et al (2020)	Shivam (2019)	Projeto atual
Acurácia	96.67%	89.47%	96.67%

Referências

SHUKLA, Asmita et al. Flower classification using supervised learning. **Int. J. Eng. Res**, v. 9, n. 05, p. 757-762, 2020.

VATSHAYAN, Shivam. Performance evaluation of supervised learning for iris flower species. 2019.