

Practical Assignment
CC2008- Machine Learning I (2024/2025)

PL5_G7

Ana Duarte Amorim - up202207213
Mariana de Sousa Serralheiro - up202208926

SUMÁRIO EXECUTIVO – FASE 1

OBJETIVO

Avaliar o comportamento do algoritmo K-Nearest Neighbors (KNN) perante problemas de noise e outliers nos datasets.

ABORDAGEM

Implementação do KNN.

Introdução de noise nas classes e outliers nas características.

Avaliação dos datasets usando várias métricas.

RESULTADOS

O K-Nearest Neighbors (KNN) é altamente sensível à presença de noise e outliers, com uma diminuição no desempenho à medida que estes problemas são introduzidos nos dados.

K-NEAREST NEIGHBOR E O PROBLEMA DO NOISE E OUTLIERS

Descrição

O algoritmo que escolhemos para este estudo foi o **K-Nearest Neighbors (KNN)**.

O **KNN** é um algoritmo de classificação de **Machine Learning** que classifica um novo exemplo, com base nas classes dos seus *k vizinhos mais próximos*. Para determinar a distância entre os exemplos, utilizamos a **distância Euclidiana**. Definimos o número de vizinhos a analisar (*k*) como 5.

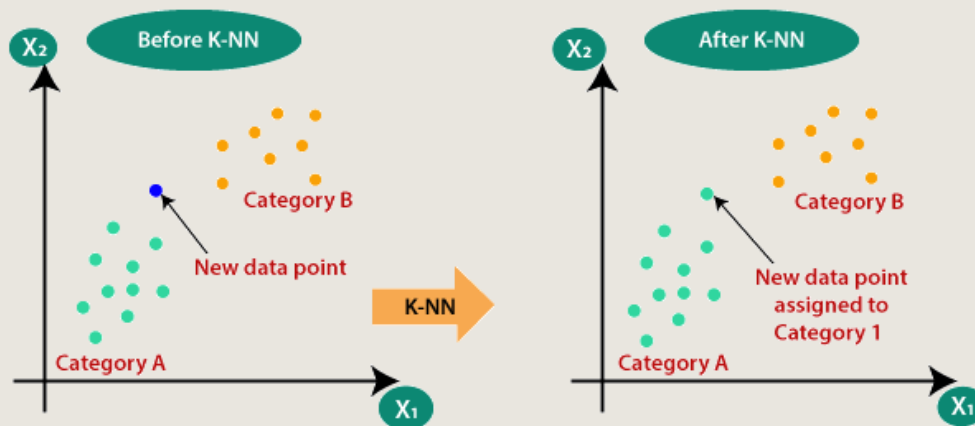
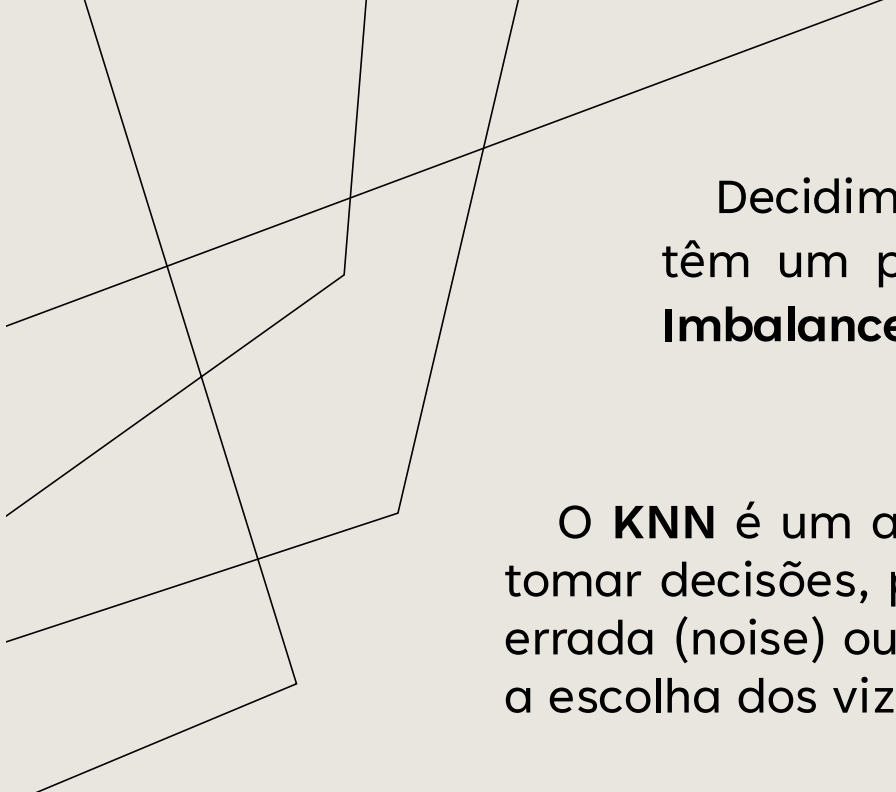


Fig.1 Algoritmo K-NN



Decidimos estudar o impacto do **Noise** e **Outliers** porque estes problemas têm um pior impacto no algoritmo KNN do que os problemas de **Class Imbalance** ou **Multiclass Classification**.

O **KNN** é um algoritmo que depende muito da distância entre os exemplos para tomar decisões, por isso qualquer alteração seja através de um dado com a classe errada (noise) ou com um valor extremo (outlier) pode afetar de forma significativa a escolha dos vizinhos e consequentemente a classificação final.

O problema de **Class Imbalance** também afeta este algoritmo mas não de forma tão imediata, porque o algoritmo consegue funcionar de forma razoável se os vizinhos mais próximos dos exemplos de teste tiverem as classes minoritárias.

Por outro o lado, o KNN adapta-se ao problema da **Multiclass Classification**, pois vai à procura da maioria entre os vizinhos, independentemente de existirem duas ou mais classes possíveis no dataset.

Comportamento do Algoritmo perante os problemas de Noise e Outliers

Para estudar o impacto do Noise e Outliers no algoritmo K-Nearest Neighbor criamos funções que introduzissem estes problemas aos datasets dados. Assim, quando damos o dataset escolhido ao algoritmo, ele questiona, sobre os níveis de noise e outliers que queremos introduzir nos datasets, e posteriormente, retorna os resultados das métricas de avaliação para esses datasets.

Ao analisarmos o impacto do nível de noise ou a quantidade de outliers introduzidos nos datasets, observámos que:

- as **métricas de avaliação tendem a diminuir** com o **aumento do noise e dos outliers**.
- mesmo adicionando um pequeno nível de ruído é suficiente para provocar alterações nas métricas.
- enquanto que, no caso dos outliers, é geralmente necessário introduzir uma maior quantidade de outliers para que se note o impacto.

Isto acontece porque o noise afeta diretamente as classes, enquanto que os outliers apenas alteram as distâncias entre os exemplos, dificultando a identificação correta dos vizinhos mais próximos, sendo necessária uma maior quantidade para afetar estes vizinhos de forma significativa.

Para comparar os resultados dos valores das métricas de classificação entre o dataset original e os datasets com a introdução de noise e ruído. Optamos por escolher **níveis de noise e ruído de 30%**.

No geral, verificamos que os datasets originais apresentam melhores valores nas métricas de avaliação, como podemos observar pelo gráfico de comparação das métricas do dataset *451_Irish*. Este resultado confirma que o K-Nearest Neighbor, é muito sensível a noise e outliers.

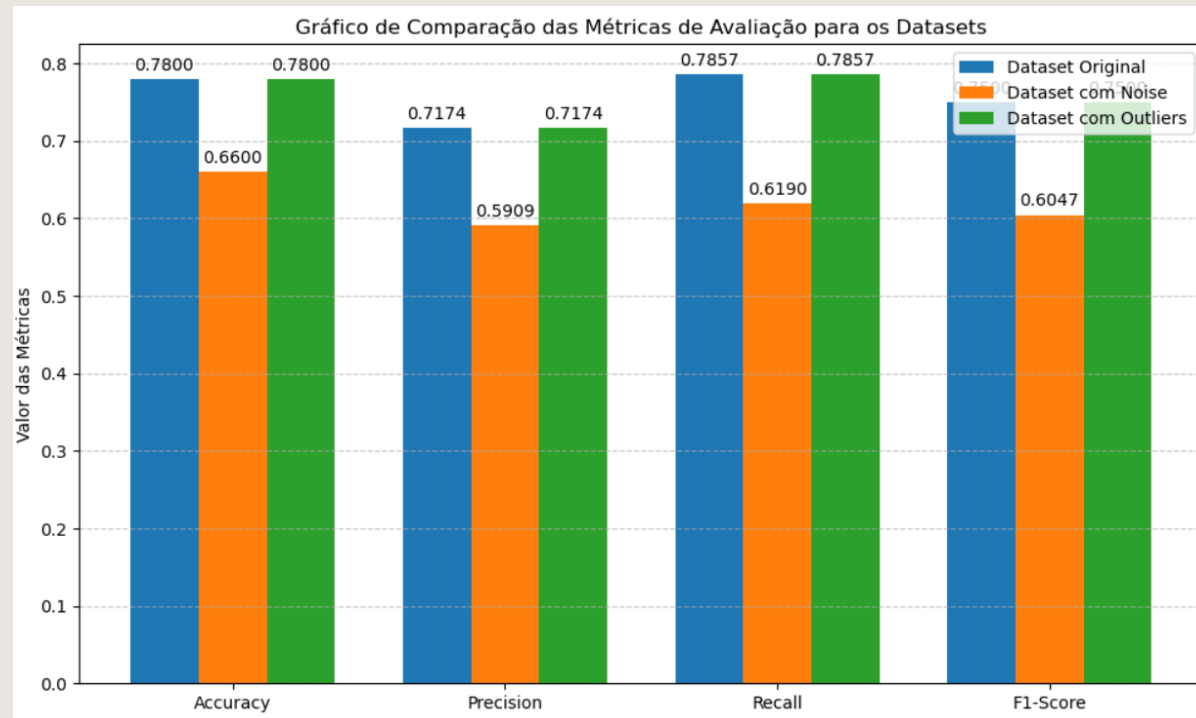


Fig.2 Gráfico de Comparação de Métricas de Avaliação para o Dataset 451_Irish

Curiosamente, em alguns casos específicos como é o caso do dataset do 50_tic-tac-toe, os datasets com outliers obtiveram métricas ligeiramente superiores ao dataset original.

Isto poderá ser explicado por:

- alguns outliers, como são criados aleatoriamente, acabaram por reforçar a separação entre as classes.
- se o dataset já continha problemas de noise ou outliers, adicionar mais dados fez com que os erros antigos tivessem menos peso

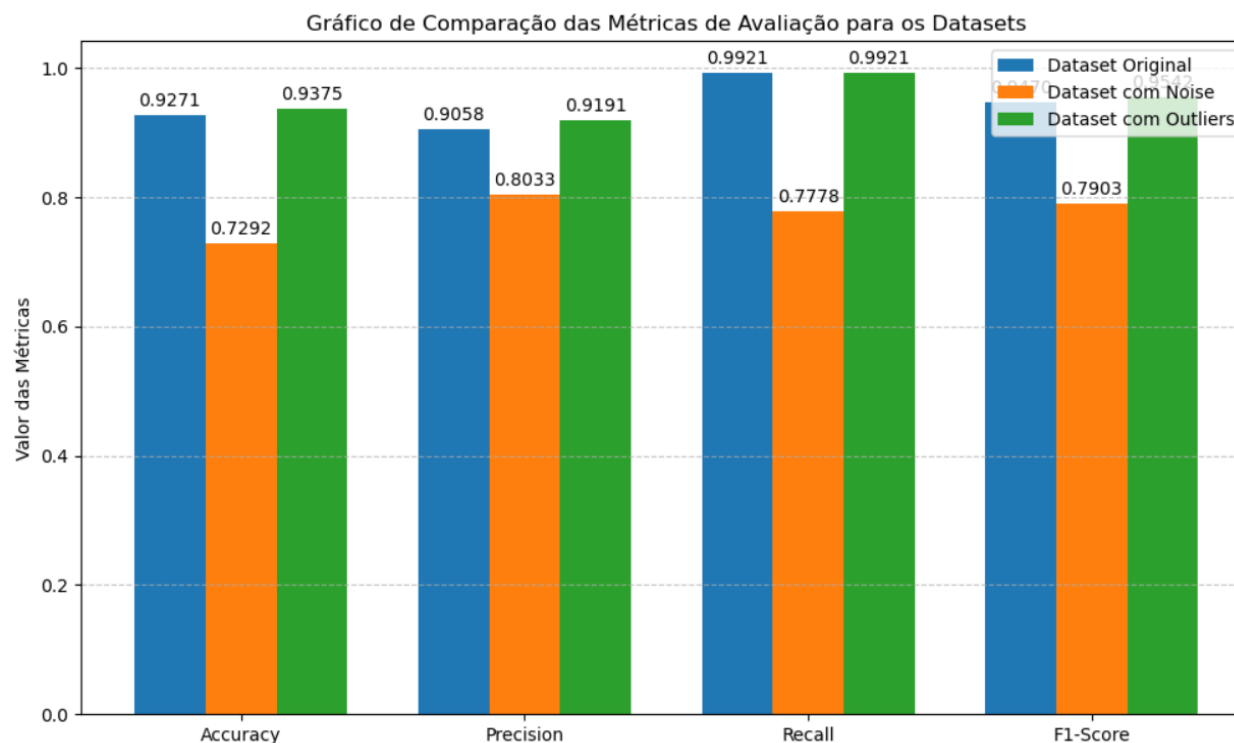


Fig.3 Gráfico de Comparação de Métricas de Avaliação para o Dataset 50_tic-tac-toe

Também observamos que em alguns datasets, a introdução de uma pequena quantidade de noise melhorou ligeiramente o desempenho do modelo, como podemos observar pelo gráfico do dataset 738_pharynx.

Pequenas quantidades de noise atuaram como uma forma de regularização espontânea, uma vez que quebraram padrões artificiais nos dados de treino.

Isto forçou o algoritmo a não depender de relações demasiado específicas, tornando-o **menos propenso a overfitting e mais capaz de generalizar**.

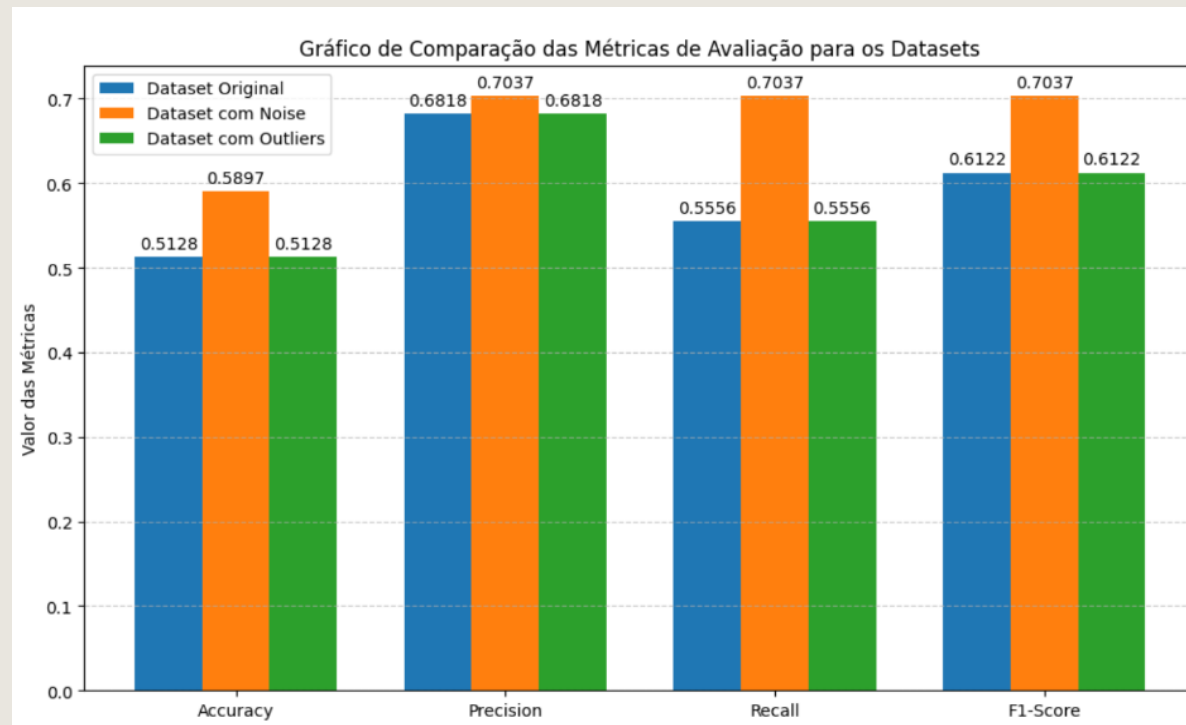


Fig.4 Gráfico de Comparação de Métricas de Avaliação para o Dataset 738_pharynx



No geral:

- O **KNN** é um **algoritmo simples** mas extremamente dependente da qualidade dos dados
- O **noise tem um impacto variável**: pequenas quantidades podem ser benéficas mas níveis mais elevados diminuem muito o desempenho
- Os **outliers têm um impacto quase sempre negativo**, perturbando os datasets e levando a classificações erradas

A sensibilidade do KNN a estes problemas reforça a importância de estratégias de pre-processamento dos dados, como deteção e remoção de outliers, ou o uso de algoritmos mais robustos ao noise.

PROPOSAL

Motivação

Durante o estudo, verificou-se que o KNN é extremamente **sensível** à presença de **noise nas classes** e de **outliers nas características**.

Pequenas perturbações nos dados são suficientes para diminuir significativamente o desempenho do algoritmo, resultando em reduções nas métricas de avaliação.

Como o KNN se baseia nas distâncias e na maioria de vizinhos para tomar decisões, não possui mecanismos para resistir a dados incorretos ou extremos.

Por isso, torna-se essencial introduzir melhorias que **aumentem a robustez do KNN** em cenários com dados ruidosos e contaminados por outliers.

PROPOSAL

Descrição

Algumas propostas de melhoria para aumentar a robustez do KNN são:

- **Aumentar o número de vizinhos (k):** definir k como aproximadamente a raiz quadrada do número de exemplos de treino (\sqrt{n}) para suavizar a influência de exemplos individuais.
- **Dar mais peso a vizinhos mais próximos,** reduzindo o impacto de outliers mais afastados.
- **Aplicar um pré-processamento** para remover exemplos que sejam outliers antes do treino.