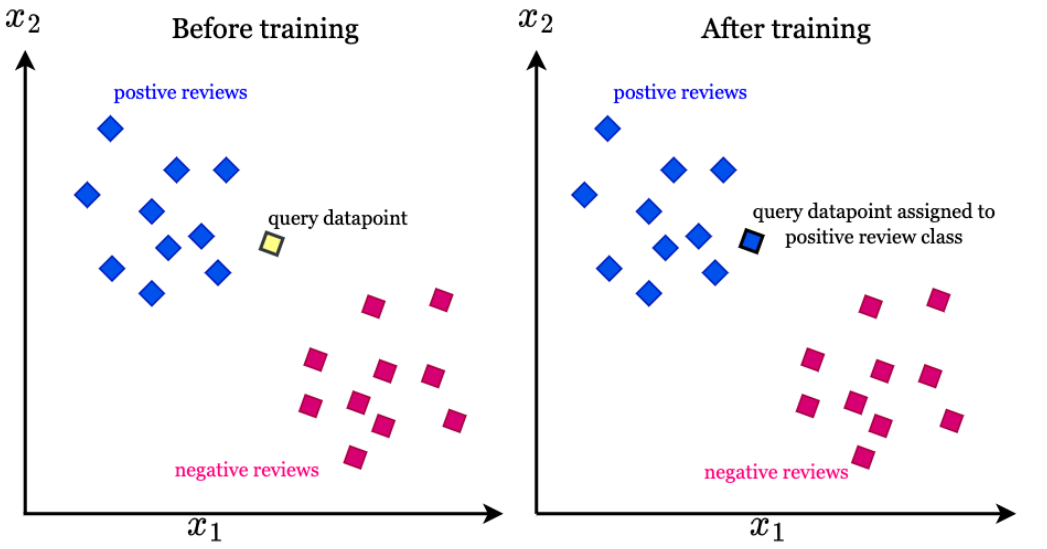
Relatório 10 - Prática: Lidando com Dados do Mundo Real (II)

Thassiana Camilia Amorim Muller

K-Nearest Neighbors (KNN)

O KNN é um algoritmo de aprendizado de máquina supervisionado que pode ser utilizado tanto para classificação quanto para regressão. Ele classifica um dado ponto com base em quão próximo ele está dos pontos em um conjunto de treinamento. É simples de usar e não faz suposições a respeito dos dados, mas pode ser muito custoso a depender do tamanho e complexidade dos dados.



(1)

O funcionamento do algoritmo se baseia, principalmente, em duas métricas:

K-valor: Quantidade de vizinhos mais próximos que o algoritmo considera ao classificar um ponto. Um valor grande de K geralmente é mais estável e sensível às variações e um valor pequeno de K é mais sensível aos dados de treinamento.

Distância entre os pontos: Algumas distâncias mais comuns usadas são a Euclidiana, Manhattan, e Minkowski. O cálculo pode impactar na performance do algoritmo.

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$ Manhattan $\sum_{i=1}^{k}|x_i - y_i|$ Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

Quando usado para classificação, o algoritmo encontra os K-pontos de treinamento mais próximos e atribui a classe mais comum entre esses vizinhos ao ponto desconhecido.

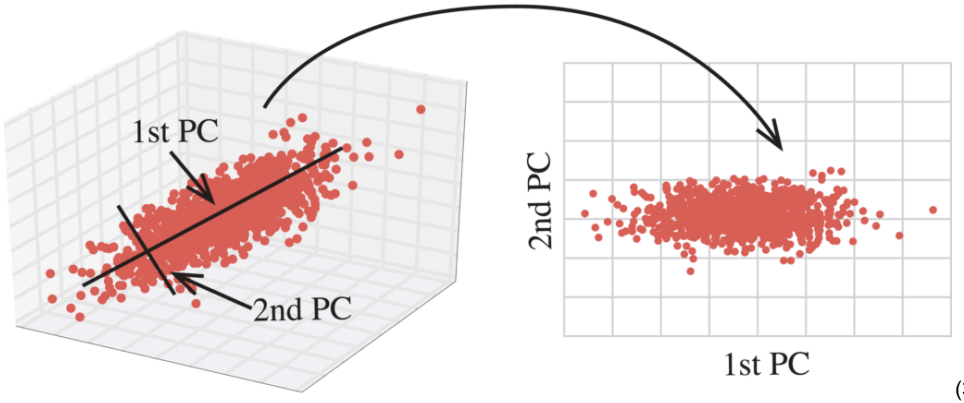
Já para a regressão, o algoritmo calcula a média (ou outra técnica) dos valores dos K-pontos mais próximos.

É muito importante normalizar os dados pois o KNN é sensível às magnitudes dos dados.
Algumas aplicações:

- Reconhecimento de padrões;
- Detecção de anomalias;
- Sistemas de recomendação.

Redução de dimensionalidade

A técnica de redução de dimensionalidade é utilizada para diminuir o número de variáveis (ou dimensões) em um conjunto de dados, mantendo o máximo possível da informação relevante. Isso é útil nos problemas de aprendizado de máquina e análise de dados pois um grande número de variáveis pode levar a modelos muito complexos, *overfitting* e alto custo computacional.



(3)

Algumas aplicações:

- Compressão de imagens;
- Reconhecimento facial;
- Análise de dados genéticos.

Data Warehousing - ETL e ELT

Data Warehousing é o armazenamento centralizado de grandes volumes de dados de diversas fontes. Esses dados são organizados para análise e relatórios. Um data warehouse é projetado para consulta e análise rápida, diferente de bancos de dados transacionais, que são otimizados para inserção e atualização de dados, geralmente são desnormalizados e possuem muitas views, funções e triggers para otimizar as consultas.

O processo para carregar os dados podem ser:

- ETL (extract, transform and load): é mais adequado para sistemas de data warehousing tradicionais onde os dados precisam ser limpos e transformados antes de serem carregados;

- ELT(extract, load and transform): é mais apropriado para arquiteturas de Big Data e data lakes onde os dados podem ser carregados de forma bruta e transformados conforme necessário.

Algumas Aplicações:

- Fornecer uma visão dos dados empresariais.
- Auxiliar nas decisões estratégicas e operacionais.
- Relatórios e Dashboards

Aprendizado por reforço

Aprendizado por reforço é uma subárea da inteligência artificial que se concentra em fornecer a agentes (robôs) alguma recompensa cumulativa de forma a ensiná-los a tomar decisões em um determinado ambiente.

Algumas aplicações:

- Jogos;
- Robótica;
- Sistemas de Recomendação;
- Finanças.

  Um exemplo prático e visual: Inteligência Artificial brincando de Pique-Esconde

Matriz de confusão

Uma matriz de confusão é uma tabela que permite visualizar o desempenho do algoritmo de classificação comparando os valores previstos com os valores reais, geralmente organizada da seguinte maneira para um problema de classificação binária:

Binary confusion matrix

	Actual YES	Actual NO
Predicted YES	TRUE POSITIVES	FALSE POSITIVES
Predicted NO	FALSE NEGATIVES	TRUE NEGATIVE

- True Positive (TP): O número de casos positivos corretamente classificados pelo modelo.
- False Positive (FP): O número de casos negativos incorretamente classificados como positivos pelo modelo.
- True Negative (TN): O número de casos negativos corretamente classificados pelo modelo.
- False Negative (FN): O número de casos positivos incorretamente classificados como negativos pelo modelo.

A partir dessa matriz, é possível obter algumas métricas de desempenho, como:

- **Acurácia:** mede a proporção de previsões corretas em relação ao total de previsões realizadas, porém pode ser enganosa em conjuntos de dados onde uma classe é muito mais frequente que outra (desbalanceados);

$$\frac{Previsões\ Corretas}{Total\ de\ Previsões}$$

- **Precisão:** A precisão mede a proporção de verdadeiros positivos entre as previsões positivas;

$$\frac{TRUE\ POSITIVES}{TRUE\ POSITIVES+FALSE\ POSITIVES}$$

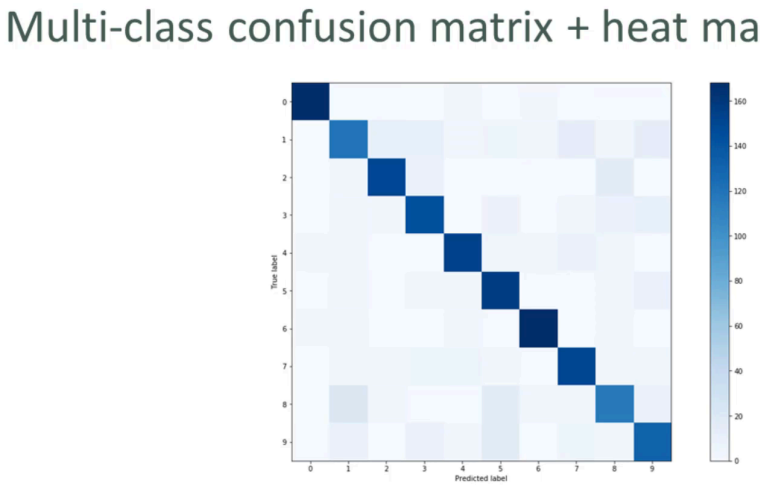
- **Recall (Sensibilidade ou Revocação)**: A sensibilidade ou revocação mede a proporção de verdadeiros positivos identificados corretamente em relação ao total de positivos reais;

$$\frac{TRUE\ POSITIVES}{TRUE\ POSITIVES+FALSE\ NEGATIVES}$$

- **F1 Score:** O F1 Score é a média harmônica entre precisão e sensibilidade.

$$2 \cdot \frac{Precision \cdot Recall}{Precision+Recall}$$

Para problemas de classificação multiclasse, a matriz de confusão se expande formando uma matriz N×N, onde N é o número de classes. Cada célula da matriz mostra a contagem de instâncias que pertencem à classe real i e são previstas como classe j.
A matriz de confusão é uma ferramenta essencial para compreender melhor o desempenho de um modelo de classificação e identificar onde ele está cometendo erros.



ROC (Receiver Operating Characteristic)

A curva ROC é uma ferramenta gráfica usada para avaliar o desempenho de um modelo de classificação binária, variando o limiar de decisão.

Ela é traçada como uma curva em um gráfico na qual o eixo X é a taxa de falsos positivos (False Positive Rate - FPR) e o eixo Y é a taxa de positivos verdadeiros (True Positive Rate - TPR)

AUC (Area Under the Curve)

AUC refere-se à área sob a curva ROC. A AUC fornece uma única métrica que resume a performance do modelo ao longo de todos os possíveis limiares de classificação.

- AUC = 1: Indica um modelo perfeito que separa perfeitamente as classes.
- AUC = 0.5: Indica um modelo que não tem habilidade discriminativa, equivalente a uma classificação aleatória.
- AUC < 0.5: Indica um modelo que está pior do que a classificação aleatória (embora isso possa significar que as previsões estão invertidas).

Bias e Variância

O *bias* ou viés é o quão longe a média dos seus dados está do valor desejado, quando muito alto não captura a complexidade subjacente dos dados causando um underfitting.

A variância é o erro introduzido pela sensibilidade do modelo às pequenas variações nos dados de treinamento. Um modelo com alta variância é muito complexo e se ajusta muito bem aos dados de treinamento, podendo causar um overfitting.

K-fold cross-validation

K-fold cross-validation é uma técnica de validação usada para avaliar o desempenho de um modelo de aprendizado de máquina e garantir que ele se generalize bem para dados não vistos. O conjunto de dados é dividido em subconjuntos menores (folds), em cada iteração, um subconjunto é usado como conjunto de validação e os outros são usados como conjunto de treinamento. Detalhes no arquivo “K_cross_aula.ipynb”

Importância de um bom tratamento e limpeza dos dados

A limpeza, tratamento e normalização não só melhoram a qualidade dos dados, mas também garantem que os modelos treinados sejam precisos, eficientes e capazes de generalizar bem para novos dados. Dados sujos podem introduzir viés nos modelos, levando a previsões incorretas ou injustas.

Os principais tipos de limpezas de dados incluem:

- Redução de Ruído: como valores extremos, entradas incorretas ou dados inconsistentes.;
- Correção de Erros: como entradas duplicadas, valores ausentes ou entradas incorretas;
- Formato Adequado: garantir que eles estejam prontos para serem alimentados nos modelos.

- Normalizar os dados: garante que todas as características irão contribuir de forma equilibrada.

Dados desbalanceados

Um conjunto de dados desbalanceado pode levar a modelos que têm um desempenho ruim em classes minoritárias, já que os algoritmos tendem a ser influenciados pelas classes majoritárias.
As principais técnicas para lidar com esse desafio são:

- Oversampling: aumenta o número de amostras da classe minoritária, pode ser feito replicando amostras existentes ou criando novas amostras sintéticas(SMOTE).
- Undersampling: diminui o número de amostras da classe majoritária.

Outras técnicas para pré-processamento dos dados

- Binning: Transforma variáveis contínuas em categóricas.
- Transforming: Modifica a distribuição dos dados.
- Encoding: Converte variáveis categóricas em numéricas.
- Scaling: Ajusta a escala dos dados.
- Shuffling: Embaralha os dados para garantir uma distribuição aleatória.

Conclusão

Assim, conclui-se que trabalhar com ciência de dados e ML na prática pode ser frustrante quando comparado aos casos estudos se o profissional não souber limpar e tratar corretamente os dados para, então, utilizá-los nas técnicas adequadas para cada tipo de problema.

Referências

H, R. S. K-Nearest Neighbors Algorithm. Disponível em: <https://intuitivetutorial.com/2023/04/07/k-nearest-neighbors-algorithm/>.(1)
KNN Classification. Disponível em: <https://www.saedsayad.com/k_nearest_neighbors.htm>.(2)
GERMANOS, L. V. Redução de Dimensionalidade. Disponível em: <https://luvi01.medium.com/redu%C3%A7%C3%A3o-de-dimensionalidade-c6d714b8ae5d>.(3)