Relatório de Desempenho de Algoritmos de Classificação: KNN e Naïve Bayes

Luis Borges(47297)

December 30, 2024

1 Introdução

Este relatório apresenta a implementação e avaliação de dois algoritmos de classificação: **K-Nearest Neighbors (KNN)** e **Naïve Bayes**. Ambos os modelos foram aplicados em quatro conjuntos de dados distintos: *Iris*, *Rice*, *WDBC* e *tic-tac-toe* um conjunto adicional obtido online do jogo do galo, onde se presume que o "x" tenha jogado primeiro. O conceito alvo é "ganhar por x" (verdadeiro quando "x" tem um de 8 formas possíveis de criar um "três em linha")Clique aqui para visitar o site. Os resultados de desempenho são analisados à luz da capacidade de generalização dos modelos.

2 Implementação dos Modelos

2.1 K-Nearest Neighbors (KNN)

Existe um construtor que aceita dois parâmetros, "k" e "p" (sendo estes o n$^{\underline{o}}$ de vizinhos e o parâmetro de distância respetivamente), com valores padrões definidos. O método "fit" recebe os conjuntos "X" e "y" e converte os dados para arrays. A distância euclidiana é calculada pela função "minkowski". A função "predict" aplica o modelo e devolve as etiquetas previstas para o conjunto fornecido à função, onde depois a função "score" vai averiguar a exatidão do modelo. O algoritmo KNN foi implementado utilizando as seguintes características:

- **Estrutura de Dados**: Os dados de treino foram armazenados como arrays Numpy para permitir cálculos eficientes.
- **Distância de Minkowski**: Foi utilizada como métrica de similaridade, parametrizada por p.
- **Hiperparâmetros**: $k \in \{1, 5, 9\}$ e $p \in \{1, 2\}$.

2.2 Naïve Bayes

Existe um construtor que aceita um parâmetro "suave" (valor padrão 1e-9) para evitar probabilidades nulas. O método "fit" recebe os conjuntos "X" e "y", calcula as médias e variâncias para atributos numéricos e as probabilidades para atributos categóricos, além das probabilidades a priori das classes. A função "predict" aplica o modelo e devolve

as etiquetas previstas para o conjunto fornecido. A função “score” avalia a exatidão do modelo comparando as predições com os rótulos verdadeiros. O classificador Naïve Bayes foi implementado considerando:

- **Distribuição Gaussiana**: Os atributos foram assumidos como seguindo uma distribuição normal.
- **Suavização**: Parâmetro de suavização $smooth \in \{10^{-9}, 10^{-5}\}$ foi adicionado para evitar divisões por zero.
- **Estrutura de Dados**: As médias e variâncias de cada atributo por classe foram armazenadas como dicionários Python.

3 Seleção dos Conjuntos de Dados

Quatro conjuntos de dados foram utilizados neste trabalho:

1. **Iris**: Disponível na biblioteca UCI Machine Learning Repository.
2. **Rice**: Dados relacionados à classificação de grãos de arroz.
3. **WDBC**: Dados de diagnóstico de câncer de mama.
4. **Tic-tac-toe**: Dados relacionados com o jogo de galo, quando o jogador "x" tem 8 formas de de criar um 3 em linha.

Os conjuntos foram divididos em 75% para treino e 25% para teste.

4 Resultados de Desempenho

Os modelos foram avaliados em termos de exatidão. A Tabela 1 apresenta os resultados obtidos:

Dataset	Algoritmo	Hiperparâmetros	Exatidão
Iris	KNN	$k = 1, p = 1$	0.9211
Iris	KNN	$k = 1, p = 2$	0.8947
Iris	KNN	$k = 5, p = 1$	0.9211
Iris	KNN	$k = 5, p = 2$	0.9474
Iris	KNN	$k = 9, p = 1$	0.9211
Iris	KNN	$k = 9, p = 2$	0.9474
Rice	KNN	$k = 1, p = 1$	0.9003
Rice	KNN	$k = 1, p = 2$	0.8919
Rice	KNN	$k = 5, p = 1$	0.9161
Rice	KNN	$k = 5, p = 2$	0.9192
Rice	KNN	$k = 9, p = 1$	0.9203
Rice	KNN	$k = 9, p = 2$	0.9297
WDBC	KNN	$k = 1, p = 1$	0.9650
WDBC	KNN	$k = 1, p = 2$	0.9510
WDBC	KNN	$k = 5, p = 1$	0.9580
WDBC	KNN	$k = 5, p = 2$	0.9510
WDBC	KNN	$k = 9, p = 1$	0.9580
WDBC	KNN	$k = 9, p = 2$	0.9580
Iris	Naïve Bayes	$smooth = 10^{-9}$	0.9737
Iris	Naïve Bayes	$smooth = 10^{-5}$	0.9737
Rice	Naïve Bayes	$smooth = 10^{-9}$	0.9255
Rice	Naïve Bayes	$smooth = 10^{-5}$	0.9255
WDBC	Naïve Bayes	$smooth = 10^{-9}$	0.9371
WDBC	Naïve Bayes	$smooth = 10^{-5}$	0.9371
Tic-Tac-Toe	Naïve Bayes	$smooth = 10^{-9}$	0.7458
Tic-Tac-Toe	Naïve Bayes	$smooth = 10^{-5}$	0.7458

Table 1: Resultados de desempenho dos modelos nos diferentes datasets.

5 Discussão dos Resultados

Os resultados mostram que:

KNN

- O desempenho é geralmente bom, com melhor generalização em $k = 5$ ou $k = 9$ e ligeira vantagem para $p = 2$ (distância Euclidiana).
- Conjuntos de dados como *Iris* e *WDBC* tiveram exatidões altas ($\sim 95\%$), enquanto o conjunto *Rice* também apresentou resultados sólidos ($\sim 92\%$).
- Para $k = 1$, houve sinais de sobreajustamento, especialmente em conjuntos menores.

Naïve Bayes

- Teve excelente desempenho em conjuntos simples como *Iris* ($97,37\%$) e *Rice* ($92,55\%$), mostrando robustez e eficiência.

- Em conjuntos mais complexos, como *Tic-Tac-Toe* $(74,58\%)$, o desempenho foi mais limitado, indicando dificuldades em capturar todas as variabilidades.

6 Conclusão

O **KNN** demonstrou melhor generalização com parâmetros ajustados, enquanto o **Naïve Bayes** foi eficaz em conjuntos mais simples, mas mostrou limitações em cenários mais desafiadores.
Este trabalho demonstrou a implementação e avaliação de dois algoritmos clássicos de classificação. Embora ambos os modelos apresentem bom desempenho geral, a escolha do algoritmo ideal depende da natureza do problema e dos dados disponíveis.