Trabalho Prático 2 - Aprendizado de máquina

Leandro Freitas de Souza - 2021037902

1 Introdução

O k-Nearest Neighbors (KNN) e o KMeans são métodos fundamentais em aprendizado de máquina e análise exploratória de dados, centrados no agrupamento de indivíduos com base em suas características. Este trabalho visa estudar a aplicação desses algoritmos, utilizando como contexto um banco de dados contendo estatísticas de jogadores de basquete. A análise exploratória desses métodos será realizada para compreender como eles podem contribuir para a categorização e compreensão de padrões nos dados relacionados ao desempenho dos jogadores.

2 k-Nearest Neighbors

O KNN opera de maneira direta, iniciando com a criação de um classificador usando o banco de dados de treinamento, no caso, o "nba_treino". Em seguida, o algoritmo realiza previsões para determinar em qual grupo um novo ponto se encaixa, com base na avaliação da proximidade desse ponto em relação aos k pontos mais próximos no conjunto de treinamento. Todos os pontos do arquivo "nba_teste"foram subsequentemente avaliados e classificados, utilizando diferentes valores de k (2, 10, 50 e 100) para aprofundar a compreensão do funcionamento do algoritmo. Esse processo de experimentação com diferentes valores de k permite analisar a sensibilidade do modelo às variações de vizinhança, fornecendo informações sobre o desempenho e a robustez do classificador.

2.1 Métricas

Para avaliar a corretude e a eficácia do programa, foram implementadas 5 métricas principais para a avaliação do código. A matriz de confusão, acurácia, precisão, revocação e f1. A seguir temos a descrição e os resultados apresentados por cada uma delas.

Matriz de Confusão

Em um algoritmo k-Nearest Neighbors (KNN), a matriz de confusão é uma representação tabular que resume as previsões do modelo em relação aos resultados reais. Essa matriz é crucial para avaliar o desempenho do KNN, destacando verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos.

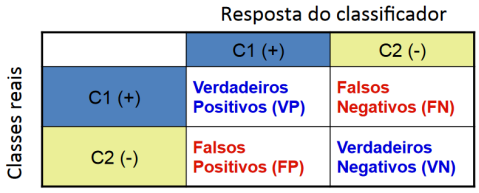

Figura 1: Representação de uma matriz de confusão

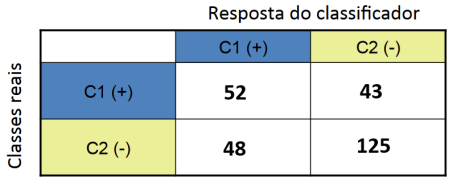

Figura 2: Matriz de confusão encontrada para k = 50

Acurácia

A acurácia representa a fração correspondente aos testes que foram corretamente classificados. Uma vez obtida a matriz de confusão, a acurácia é dada por

$$\frac{VP + VN}{VP + FP + VN + FN}$$

No código implementado, a acurácia obtida foi por volta de 65%, além de ter sido crescente em relação ao valor de k, como observado na figura 3. Tal valor mostra que mais da metade dos jogadores foi devidamente classificada em todos os casos, mas que é mais interessante para melhor acurácia escolher valores de k maiores.

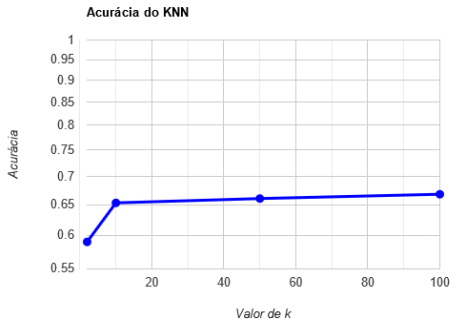

Figura 3: Acurácia do KNN

Precisão

A precisão representa a fração correspondente aos testes classificados como corretamente positivos, dentre todos os que foram classificados como positivos. Dessa forma, com a matriz de confusão podemos obter a precisão a partir da fração

$$\frac{VP}{VP + FP}$$

No código implementado, temos que a precisão média foi por volta de 73%, o que é bom, uma vez que mostra que grande parte dos valores atribuídos como positivos são corretamente atribuídos. Como podemos ver na figura 4, esse valor varia pouco, sendo maior nos extremos do gráfico, e menor em $k = 50$, onde possui valor 0,7225.

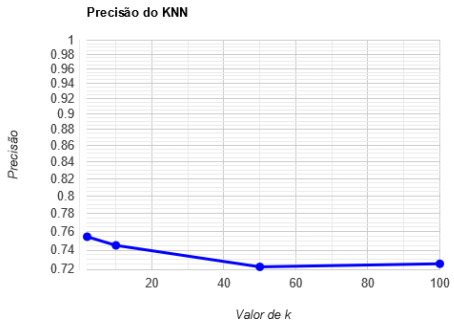

Figura 4: Precisão do KNN

Revocação

A revocação, por sua vez, representa a fração correspondente aos testes classificados como positivos, dentre os que realmente são positivos. Logo, pela matriz de confusão, podemos calcular a revocação

por meio de

$$\frac{VP}{VP + FN}$$

No código, a média para esse valor está por volta de 70%, sendo mais baixa em valores mais baixos de k, o que mostra que com menos comparações, o código atribui uma taxa maior de falsos negativos.

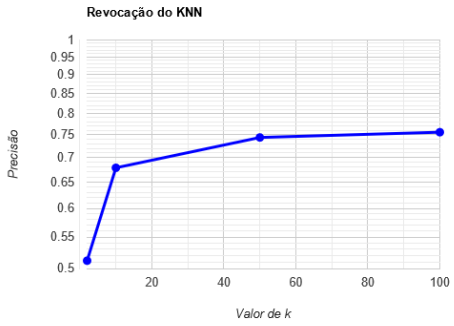

Figura 5: Revocação do KNN

F1

O F1 é um índice que é obtido através do cálculo da média harmônica entre os valores de Revocação e de Precisão do KNN. Um valor alto para F1 significa que ambos os atributos utilizados no cálculo da média também são elevados. Assim, podemos obtê-lo a partir da seguinte fórmula:

$$\frac{2 * \text{precisão} * \text{revocação}}{\text{precisão} + \text{revocação}}$$

No código implementado, o índice de F1 se manteve próximo de 70%, sendo consideravelmente pior no caso em que $k = 2$.

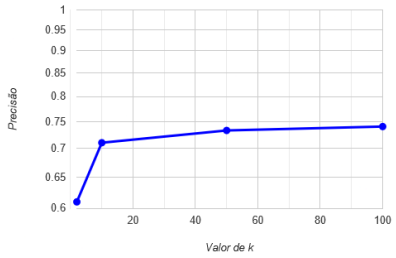

Figura 6: F1 do KNN

2.1.1 Conclusão

A partir dos resultados e dos gráficos obtidos, temos que valores muito baixos de k podem levar à atribuição de mais falsos negativos, o que os torna valores ruins para k. No entanto, valores de k maiores não surtiram grandes diferenças na eficácia da classificação.

2.2 kNN no scikit-learn

Além da implementação feita, foi utilizada, para fins de comparação, a biblioteca scikit-learn do python, que já possui esse algorítmo implementado. Para todos os valores de k, o algorítmo da biblioteca apresentou exatamente os mesmos resultados e gráficos, o que torna desnecessário a sua apresentação nesse documento. Isso mostra a efetividade e a corretude do algorítmo implementado.

3 KMeans

Uma vez que o KMeans é um tipo de aprendizado não supervisionado, as tabelas "nba_teste"e "nba_treino"foram combinadas em uma só, com nome "nba.csv". O algorítmo KMeans consiste de apenas três passos: definir k centróides, identificar os centróides mais pertos de cada ponto, e redefinir os centróides como o ponto médio entre os pontos aos quais foram atribuídos. Eventualmente o algorítmo converge para uma solução.

No caso do algorítmo implementado, foram testados dois valores de k diferentes, 2 e 3. Não é possível obter uma acurácia para $k = 3$, uma vez que os dados não foram classificados em 3 classes. Para $k = 2$, temos que a precisão obtida foi por volta de 60%, o que é bom, visto que o algorítmo não possui nenhum tipo de pré-processamento. Além disso, em todas as iterações diferentes, o algorítmo tende a convergir para os mesmos centros.

3.1 KMeans no scikit-learn

Para esse algorítmo, também foi utilizada a versão do scikit-learn para efeitos de comparação. Nesse caso, foi observada uma pequena diferença, de ordem no máximo 10^{-4} na coordenada dos centróides encontrados, o que evidencia que a diferença encontrada é mínima.

3.2 Caso em que $k = 3$

No caso em que $k = 3$, foi observada uma divergência maior do scikit-learn, porém não muito grande. No entanto, o algorítmo ainda encontrou três grupos para os jogadores, por mais que na realidade eles não existam.