Universidade Federal de Pernambuco

Aprendizagem de Máquina

Relatório da Lista 1

Aluno:	João Vitor da Silva Gomes	Data: 31/03/2019

1. Metodologia

a. Bases de dados

As bases de dados utilizadas nesta lista são provenientes do repositório *Promise Repository* (http://promise.site.uottawa.ca/SERepository/datasets-page.html). Das bases disponíveis, foram escolhidas:

CM1/*Software defect prediction*
Possui 21 atributos numéricos, e 1 categórico, que classifica o padrão quanto a presença de defeito.
A proporção por classe é:
false: 449 = 90.16%
true: 49 = 9.83%

PC1/*Software defect prediction*
Possui 21 atributos numéricos, e 1 categórico, que classifica o padrão quanto a presença de defeito.
A proporção por classe é:
false: 77 = 6.94%
true: 1032 = 93.05%

b. Modelos

Três tipos de classificadores k-NN foram construídos utilizando a linguagem de programação *Python*. Sendo eles:

k-NN clássico: Onde a classificação do padrão de teste é feito através do voto majoritário entre os k padrões de treino mais próximos.

k-NN ponderado: Que utiliza o inverso do quadrado da distância como forma de ponderar a contribuição dos k padrões de treinamento mais próximos.

k-NN adaptativo: Onde é atribuído a cada padrão de treinamento um valor de raio, a fim de informar sua proximidade com uma possível fronteira.

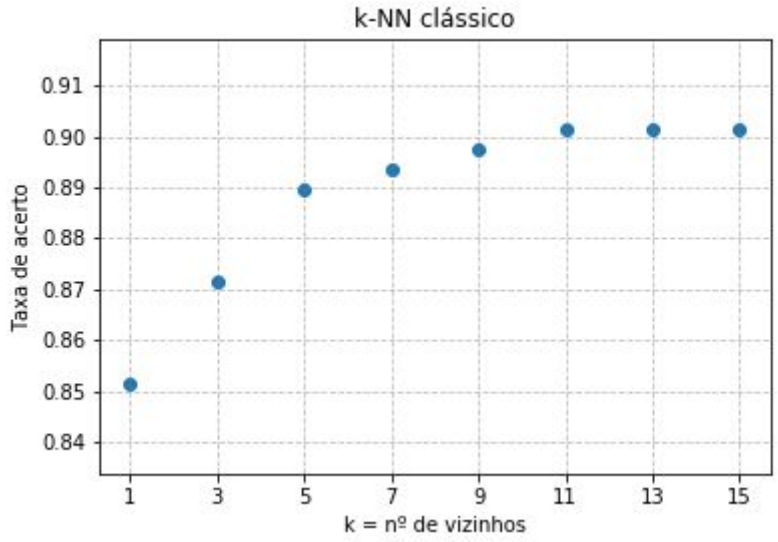
c. Avaliação

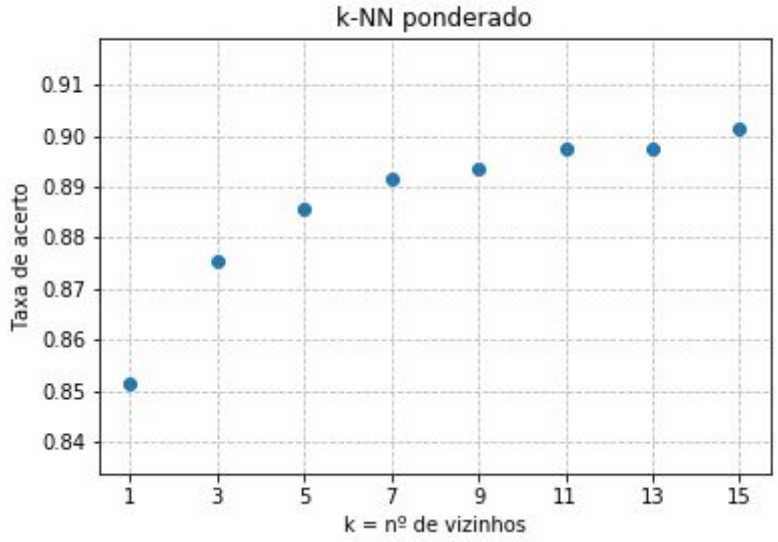
Para avaliar os modelos sobre as bases de dados, foi utilizada a técnica *stratified k-fold cross validation* com 10 seções. Entretanto, o desbalanceamento geral dos dados não foi tratado.

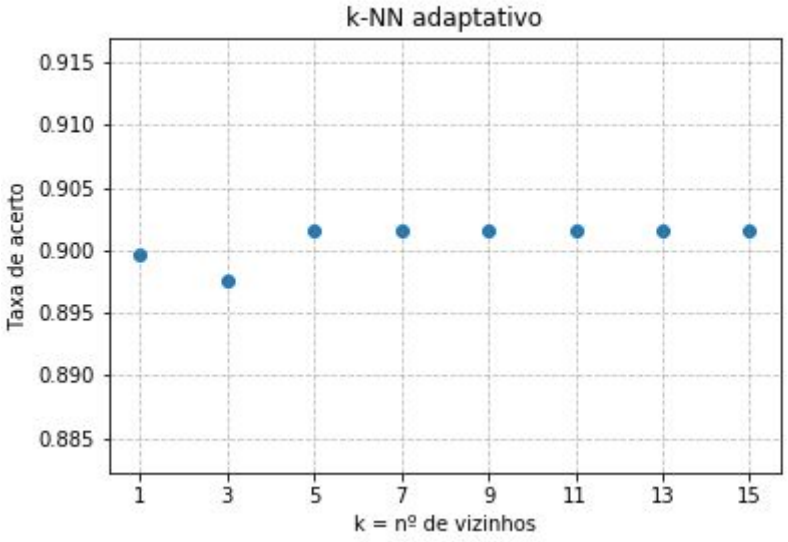
Para cada modelo, a quantidade de vizinhos próximos considerados foi variado entre valores ímpares de 1 a 15. E as métricas utilizadas para a avaliação do desempenho foram: a taxa de acerto, o tempo de treinamento e o tempo de teste.

2. Resultados

Quanto à taxa de acerto e os tempos de treinamento e teste para o conjunto de dados CM1:

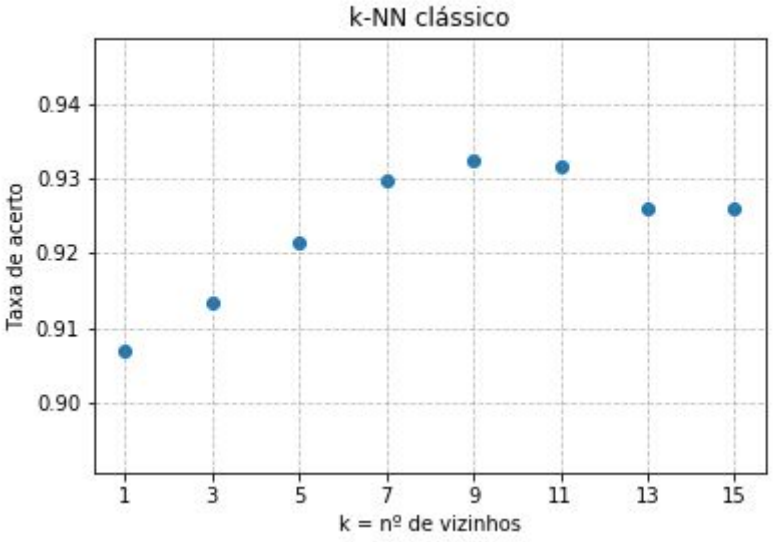


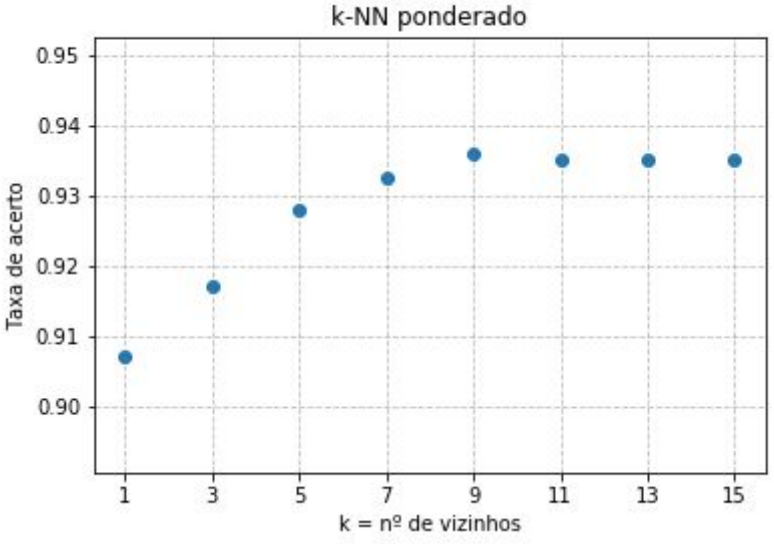


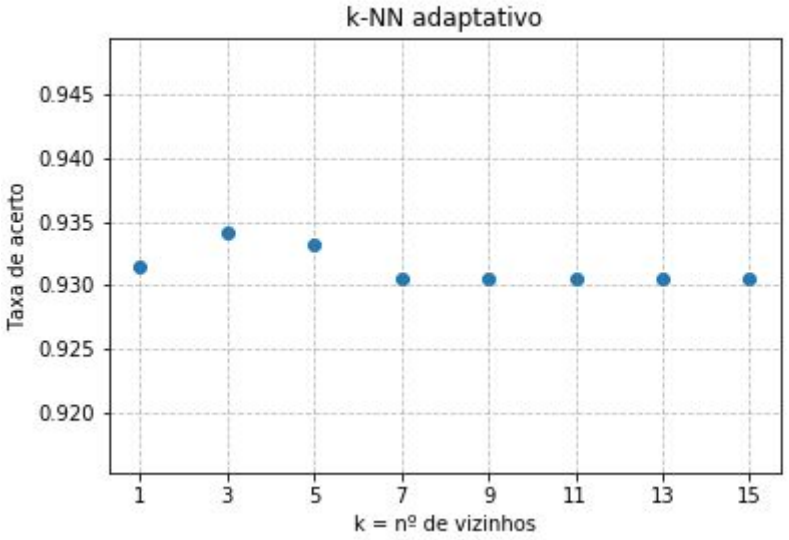


CM1	k-NN clássico	k-NN ponderado	k-NN adaptativo
Tempo de treinamento	0	0	66.4798 s
Tempo de teste	27.8626 s	29.1927 s	33.1229 s

Quanto à taxa de acerto e os tempos de treinamento e teste para o conjunto de dados PC1:







PC1	k-NN clássico	k-NN ponderado	k-NN adaptativo
Tempo de treinamento	0	0	256.5167 s
Tempo de teste	169.8677 s	149.8436 s	158.3471 s

3. Análise dos resultados

Para o conjunto de dados CM1 verifica-se que tanto para o k-NN clássico quanto para o ponderado, a taxa de acerto aumenta de forma diretamente proporcional a k. Este crescimento é acentuado entre valores de k menores que 7. Enquanto que a implementação adaptativa, exibe uma taxa que permanece constante para quase todos os valores de k, exceto entre 1 e 3, onde há um decrescimento.

Como esperado, o tempo total de processamento do k-NN adaptativo supera facilmente os tempos dos outros dois classificadores. E isso se deve ao fato de este ter um tempo de treinamento associado ao cálculo dos raios. Sendo este tempo de treinamento 2 vezes maior que o tempo gasto durante o teste.

Dependendo da aplicação, se o tempo de treinamento do k-NN adaptativo não for um problema significativo, então escolher esse modelo com k=3 é um ótima solução. Caso contrário, é melhor optar pelo modelo clássico com k=11.

Para o conjunto PC1, nos modelos clássico e ponderado, percebe-se um crescimento da taxa de acerto para até 9 vizinhos próximos. Após isto, há um decrescimento que é mais leve no modelo ponderado. Na implementação adaptativa, observa-se o mesmo comportamento de constância observado na análise para o conjunto CM1, contudo ao invés de um mínimo, há um pico em k=3.

O modelo adaptativo novamente se mostra mais custoso em relação ao tempo de execução. Aqui, todos os modelos tiveram um incremento no tempo total em relação ao conjunto CM1. Isto se deve a maior quantidade de padrões.

Para PC1, utilizar o modelo ponderado considerando os 9 vizinhos mais próximos é uma ótima escolha.