# Regressão linear múltipla como método para previsão de correções de vulnerabilidades em uma rede Heterogênea

Lucas Geraldo Cilento

March 6, 2021

## 1 Introdução

Este relatório busca descrever a metodologia utilizada e os resultados obtidos para o primeiro experimento realizado na disciplina de Aprendizagem de Máquina. O restante do documento é estruturado da seguinte forma: seção 2 mais explicações sobre a base de dados, seção 3 fornece uma breve explicação sobre os algorítimos implementados, seção 4 fornece mais detalhes referentes as métricas de avaliação, a seção 5 fornece mais detalhes sobre a execução do experimento em si, a seção 6 apresenta os resultados do experimento e por fim a seção 7 apresenta as conclusões do experimento.

## 2 Base de dados

### 2.1 Base de dados

Para este experimento foram utilizadas duas bases de dados provenientes do repositório Promise e são referentes á previsão de defeitos de software.As bases de dados selecionadas foram as seguintes: KC2/Software defect prediction KC1/Software defect prediction Os atributos que armazenam as classes destas bases são as seguintes: classes "problems" para a base kc2 e a classe "defects" para KC1. Estas classes são binárias onde o valor $True|Yes$ indica a classe positiva com a presença de defeito e a classe $False|No$ indica a classe negativa que aponta a ausência de defeito no software.

Ambas as bases possuem os mesmos atributos e nenhuma delas possui valores faltantes. As duas bases estão desbalanceadas e possuem mais instâncias com a classe negativa. A tabela abaixo mostra a distribuição das classes para cada base:

| | Classe Negativa | Classe Positiva |
|---|---|---|
| KC2 | 415 (79.5%) | 105 (20.5%) |
| KC1 | 1783 (84.55%) | 326 (15.45%) |

Table 1: Distribuição das classes por base de dados

# 3 Métodos

## 3.1 K-NN

Para este algoritmo foi implementado o k-NN padrão, que leva em consideração apenas a classe dos k vizinhos mais próximos e retorna a classe mais frequente dentre os k vizinhos.

## 3.2 k-NN com distância ponderada

A implementação deste algoritmo é similar ao k-NN padrão, mas a definição da classe predita leva em consideração a distância de cada vizinho. Cada vizinho é multiplicado por um peso $p$ que pode ser descrito da seguinte forma:

$$p = \frac{1}{d(x, x_i)^2} \tag{1}$$

Onde $x$ é a instância, $x_i$ é um vizinho e $d(x, x_i)$ é a distância euclidiana entre as instâncias.

## 3.3 k-NN com distância Adaptativa

A implementação deste algoritmo é similar ao k-NN padrão, mas ao invés de utilizar a distância euclidiana para calcular os k vizinhos mais próximos é implementado a distância Euclidiana adaptada. A distância euclidiana adaptada para uma instância $x_1$ da base de teste e $x_2$ um vizinho qualquer da base de treino é descrita na fórmula abaixo:

$$d_{new} = \frac{d(x_1, x_2)}{z} \tag{2}$$

Onde $z$ é o maior raio de uma circunferência centrada em $x_2$ que exclui todas as instâncias com classe oposta a $x_2$. Esse algorítimo prioriza os vizinhos mais próximos que estão mais afastados da zona de fronteira entre as duas classes.

# 4 Avaliação do desempenho

Para avaliar a precisão dos algorítimos testados foram utilizadas a taxa de falso positivo(TP-RATE) e a taxa de falso negativo(FP-RATE). Essas métricas são mais adequadas que a acurácia para medir o desempenho das previsões em bases de dados desbalanceadas e também são métricas recomendadas pelo fornecedor da base. As formulas que descrevem a TP-rate e FP-RATE são respectivamente:

$$TP_{rate} = \frac{TP}{N} \qquad (3) \qquad\qquad FP_{rate} = \frac{FP}{P} \qquad (4)$$

Onde TP indica a quantidade de classes positivas previstas com sucesso, FP é a quantidade de falsos positivos, N é a quantidade total de classes negativas no conjunto de teste e P é o total de classes positivas no conjunto de teste.

# 5 Treino e teste

Para o treino deste experimento foi utilizado *10-fold-cross validation*. Antes de realizar o *cross-validation* também foi realizado um *shuffle* para cada base de dados, visto que as classes positivas e negativas estava agrupadas na base. O experimento foi realizado utilizando os valores de k igual à 1,2,3,5,7,9,11,13,15 para cada um dos três algorítimos citados na seção 3. Na implementação dos algorítimos a recuperação dos k vizinhos mais próximos funciona definindo a distância da instância investigada para todos os vizinhos na base de treino e depois ordenando os vizinhos pela distância. No k-NN padrão e no k-NN com peso a distância utilizada é a distância euclidiana, já no k-NN adaptativo é utilizado a distância euclidiana Adaptada.

No treino do k-NN com distância adaptativa foi adicionado mais uma etapa. Nesta etapa é criada uma lista que armazena o maior raio que exclui todos os vizinhos da classe oposta. Esta lista é utilizada para fornecer o raio da distância euclidiana adaptada no calculo do k-NN adaptativo.

# 6 Resultados

## 6.1 Análise base KC2

A figura 1 mostra um gráfico do desempenho de cada modelo na base KC2 em uma curva ROC. Neste gráfico é visível que modelos com $K < 3$ apresentam os piores desempenhos em relação à taxa FP apresentando mais falsos positivos enquanto os classificadores com $K >= 13$ apresentam bons resultados em relação a taxa FP mas apresentam uma piora em relação à taxa TP.

A figura 2 mostra a evolução da taxa TP para os modelos avaliados de acordo com o valor de K. Neste gráfico é possível observar que o valor máximo de TP para cada algorítimo se encontra no intervalo de K entre 5 e 9. Também é

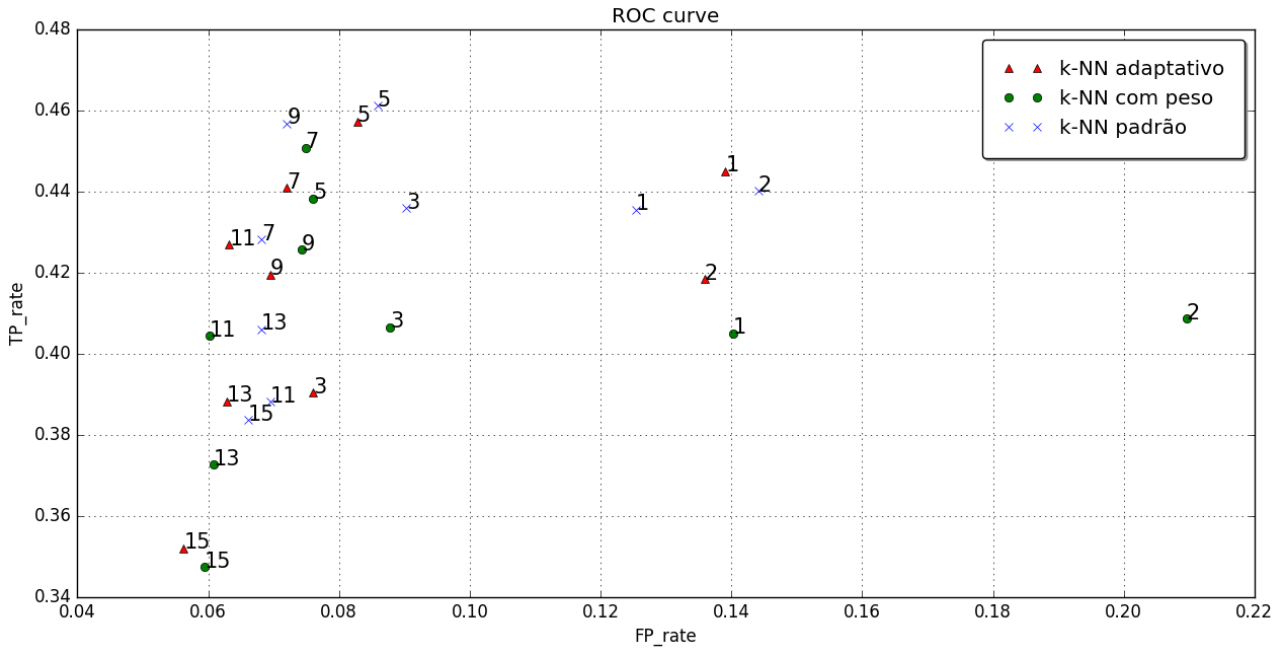


Figure 1: Curva ROC para KC2

notável a tendência de queda da taxa para valores maiores que 9. Na figura 3 temos o gráfico de evolução da taxa FP que Apresenta uma tendência de queda entre $k = 2$ e $k = 3$ e depois apresenta uma tendência decrescente atenuada até $k = 15$.

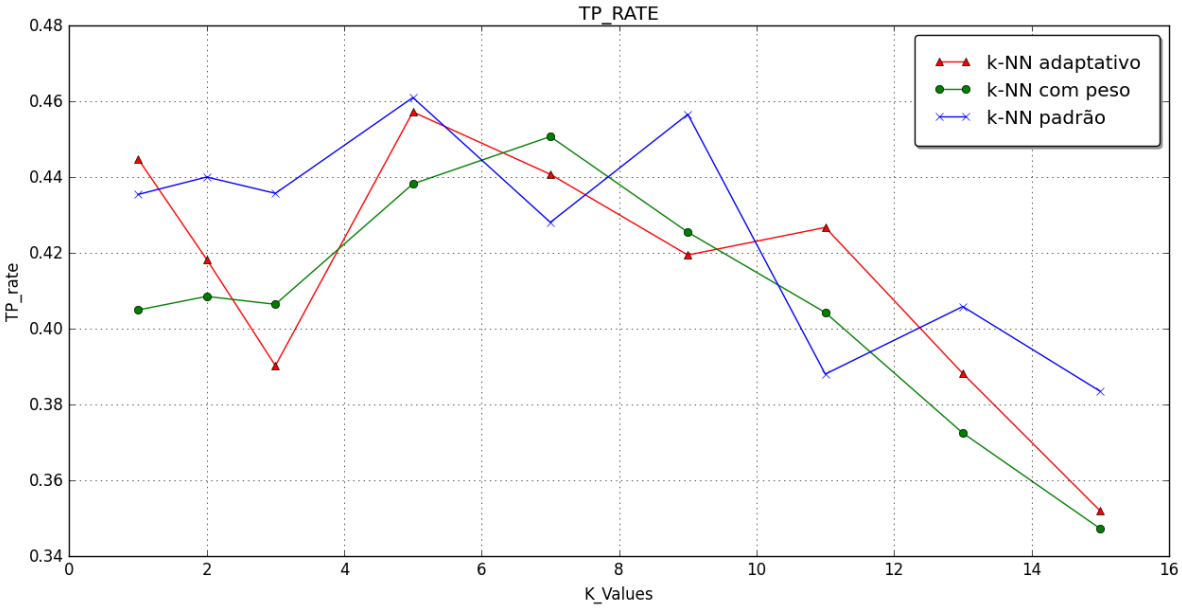


Figure 2: Evolução TP rate para KC2

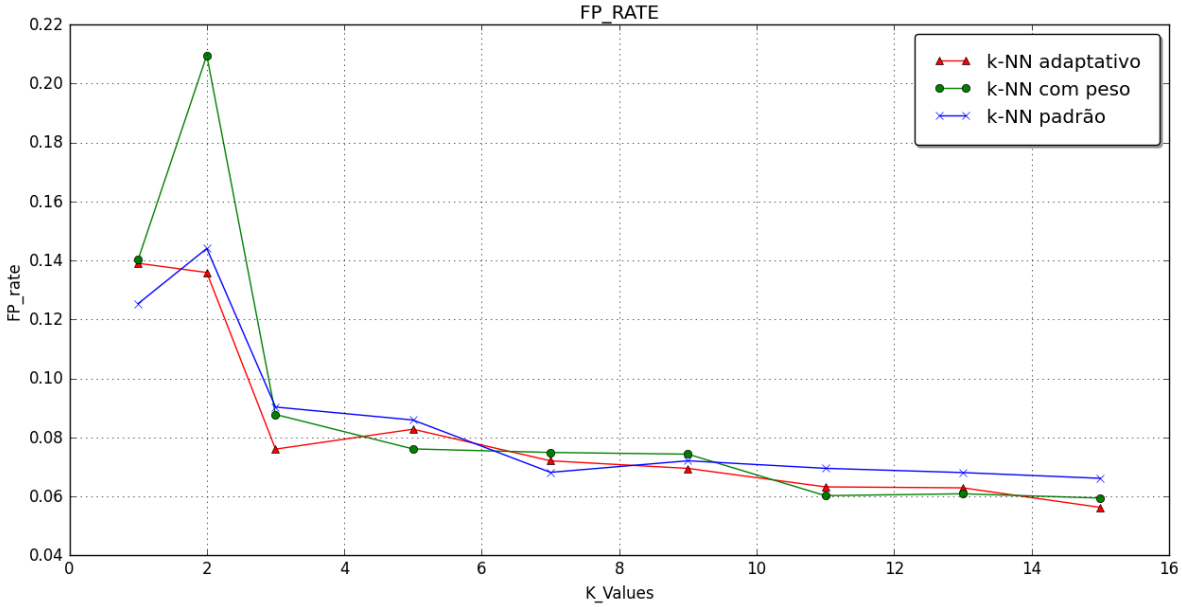


Figure 3: Evolução FP rate para KC2

## 6.2 Análise base KC1

A figura 4 mostra um gráfico do desempenho de cada modelo na base KC1 em uma curva ROC. Os desempenhos dos algorítimos se agruparam em 3 grupos: Para $K > 3$ os algorítimos apresentaram os melhores desempenhos em relação à TP rate, mas em contrapartida apresentaram os maiores valores de FP rate. Para $3 <= K <= 5$ os modelos apresentaram desempenho da TP rate menor que o grupo anterior mas também apresentram melhor desempenho em relação à FP rate. Para $K > 5$ a TP rate diminui ainda mais assim como FP rate. Na figura 5 é notável a queda do desempenho entre $K = 2$,$K = 3$ e entre $K = 5$,$K = 7$, sendo esta última diferença a mais aguda. Na figura 6 o gráfico mostra uma clara tendência de queda para FP rate conforme $K$ aumenta.

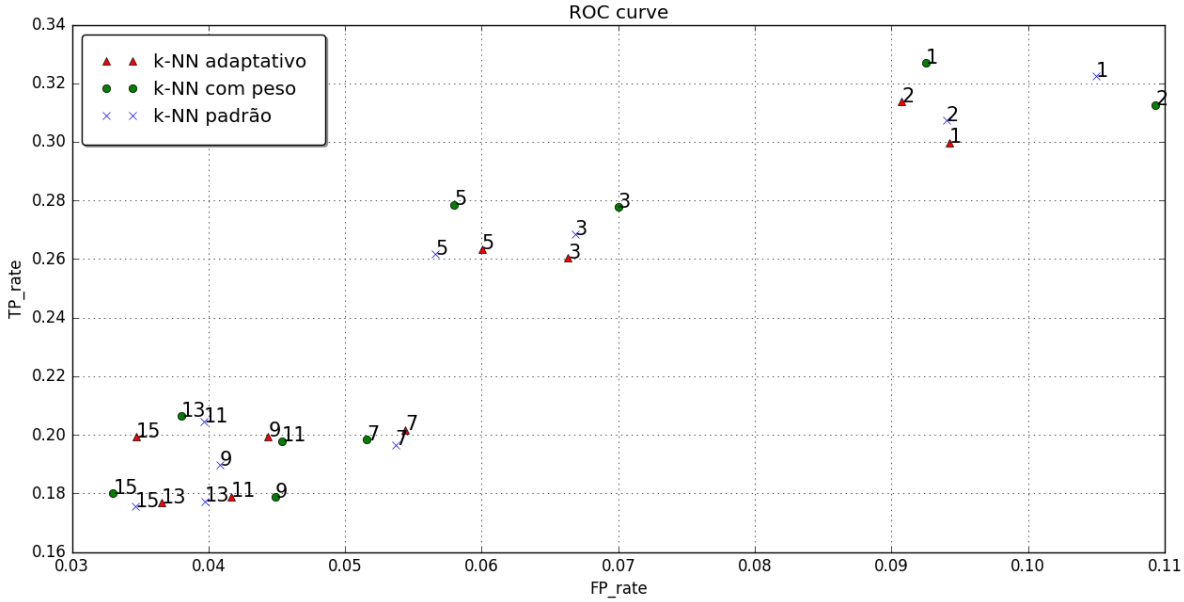


Figure 4: Curva ROC para KC1

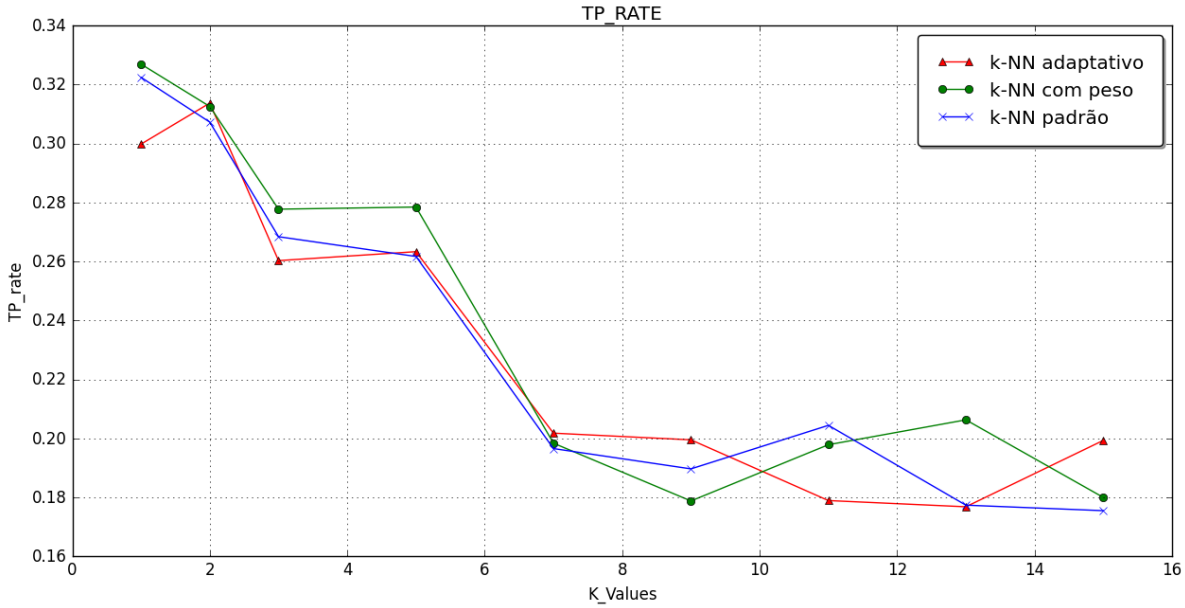


Figure 5: Evolução TP rate para KC1

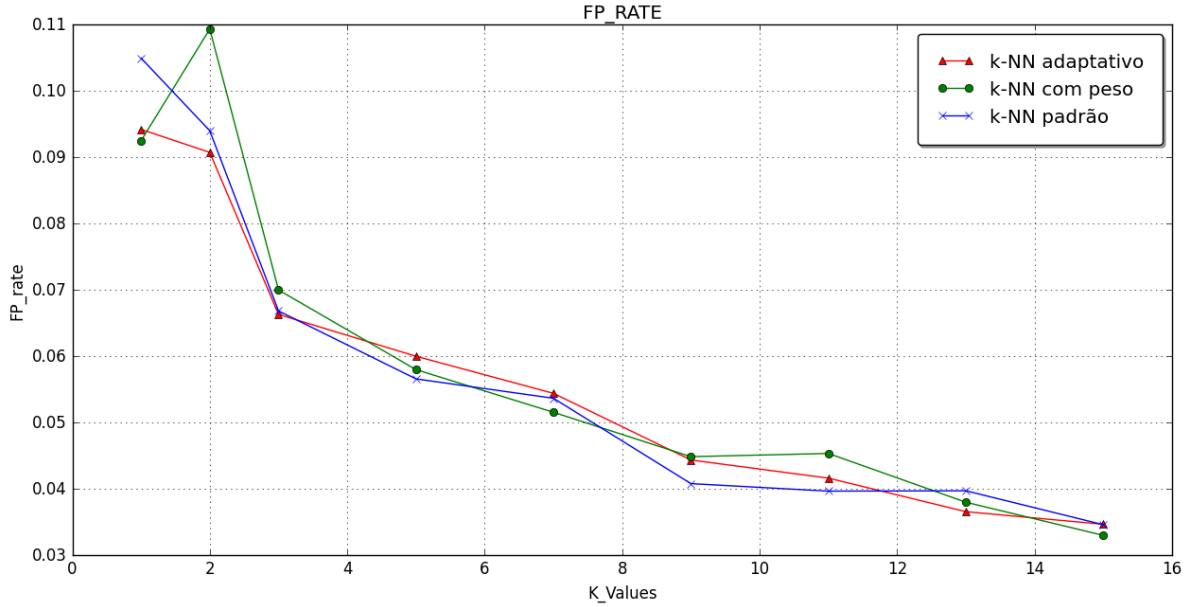


Figure 6: Evolução FP rate para KC1

# 7 Conclusão

A partir dos resultados obtidos é possível confirmar que a TP rate e a FP rate estão correlacionadas de forma que sempre que uma delas cresce a outra tende a acompanhar este crescimento. Para valores de $K$ maiores que 7 a taxa de falso positivo tende a ser bem pequena, mas em contrapartida a taxa de true positive também é pequena. Para valores de $K$ menores que 3 tempos um acréscimo da taxa de FP e da taxa TP.

Ambas as bases são referentes a avaliação de características de softwares que podem auxiliar na previsão de erros no software. Neste caso, a função dos modelos aplicada a industria é indicar a probabilidade de um determinado software possuir um erro não mapeado. Levando em consideração o contexto de criticidade onde estas bases foram geradas (bases fornecidas pela NASA) é mais importante que o modelo indique a presença de uma determinada falha, mesmo que erre na previsão, do que deixar de apontar uma falha.

No pior dos casos, quando uma falha é apontada erroneamente, será gasto tempo e dinheiro na validação dos softwares. Já no pior dos casos quando uma falha deixa de ser apontada ela pode passar despercebida por outros mecanismos de auditoria e, a depender do sistema, pode levar a uma vulnerabilidade no sistema que pode acarretar em perdas financeiras e vazamentos de informações sensíveis (incluindo segredos industriais). A falha não detectada também pode levar ao não funcionamento de um determinado sistema crítico que, em casos extremos, pode levar até a perda de vidas humanas. Por esses motivos o modelo deve ser avaliado principalmente pela taxa de acertos em relação a indicações de falhas.

Desta forma, os modelos recomendados são o 1-NN com peso para a base de dados KC1 e o 5-NN padrão para a base de dados KC2.