Relatório da atividade da disciplina exploração e mineração de dados (Técnicas)

Universidade Federal de Alagoas (UFAL)
Instituto de Computação (IC)
Exploração e Mineração de Dados

Prof. Dr. Baldoino Fonseca

Mestrando: Randerson Douglas R. Santos
rdrs@ic.ufal.br

1. Are there correlations among software metrics?

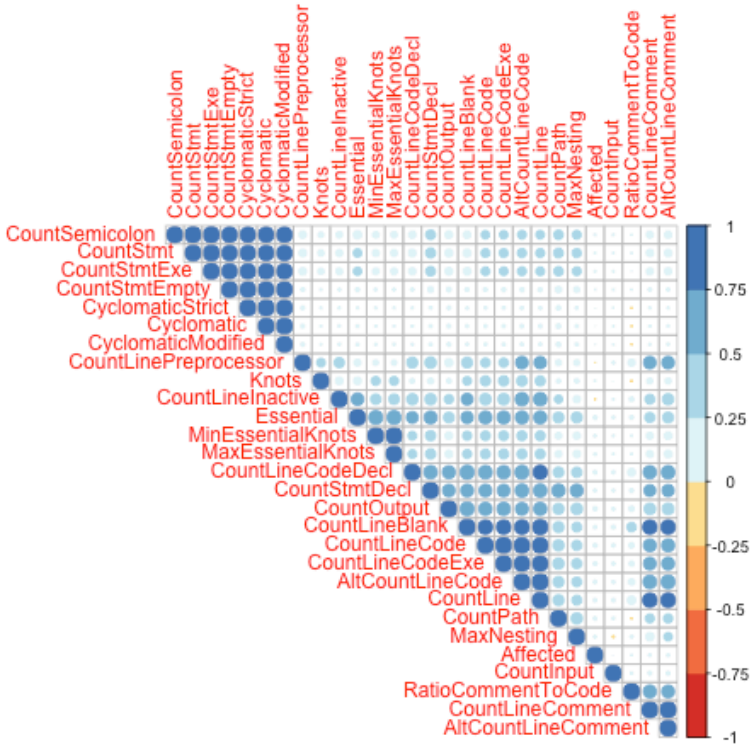
(Existem correlações entre as métricas de software?)

1.1. No estudo da aplicação da correlação entre as métricas estabelecidas adotamos as seguintes medidas para analise de dados:

- Sendo o nível de confiança: 95%.
- Sendo o nível de significância: 0.05%.

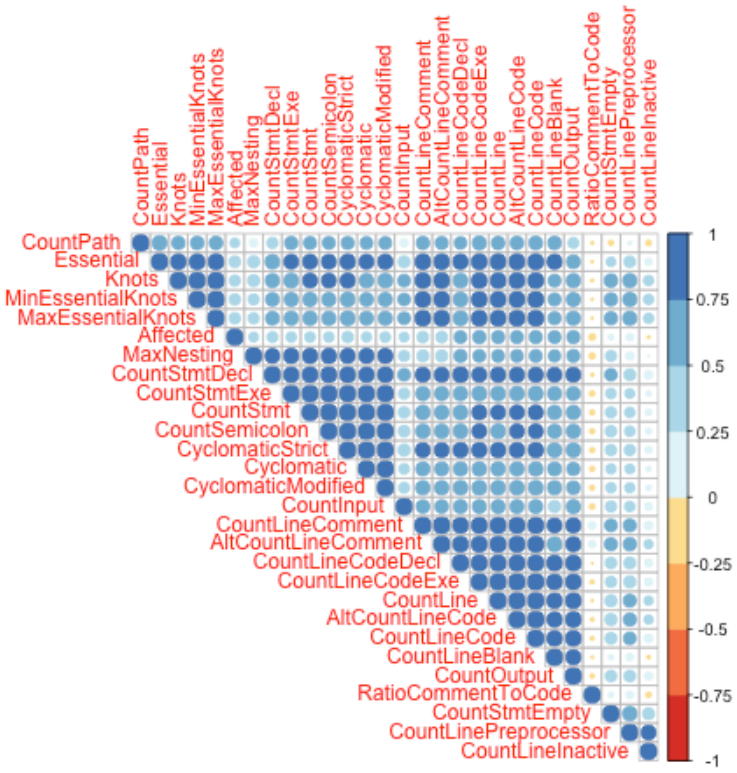
Contudo para que o entendimento fique claro, os níveis de significância estão distribuídos por cores indicando a significância ou não do valor-p (p-value), podendo ser medido de acordo com a intensidade das cores, que quando for escura indica maior significância e quando mais clara, menor significância.

1.2. Análise das bases:
DATASET GLIBC NORMAL



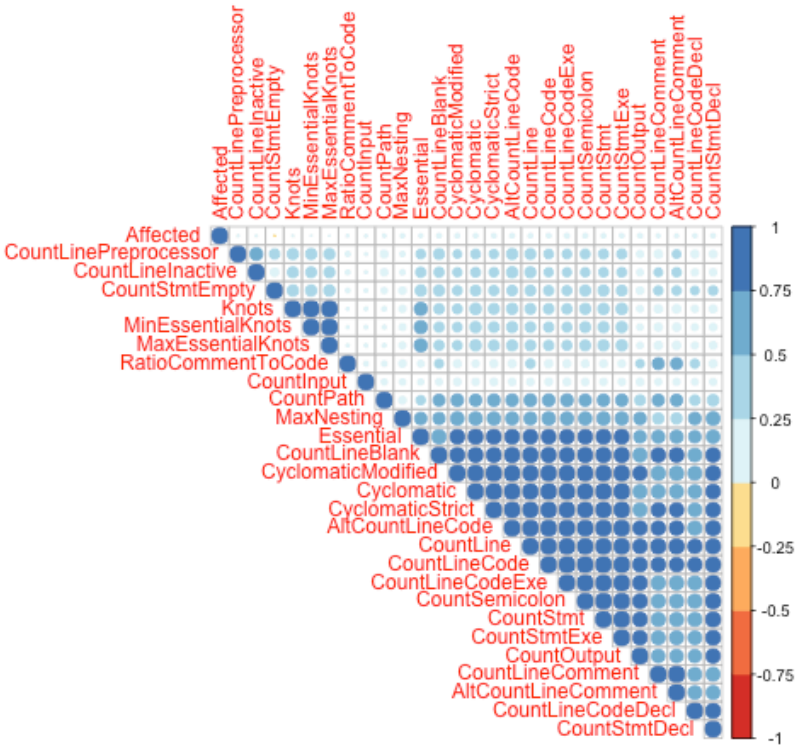
Observa-se que no dataset acima há um pequeno índice de métricas correlacionadas, mas percebe-se que há um agrupamento entre elas. Como mostra a imagem acima existe uma correlação de maior significância entre as CountSemiColon, CountStmt, CountStmtExe, CountStmtEmpty, CountLineBlank, CountLineCode, CountLineCodeExe.

DATASET GLIBC BALANCEADO



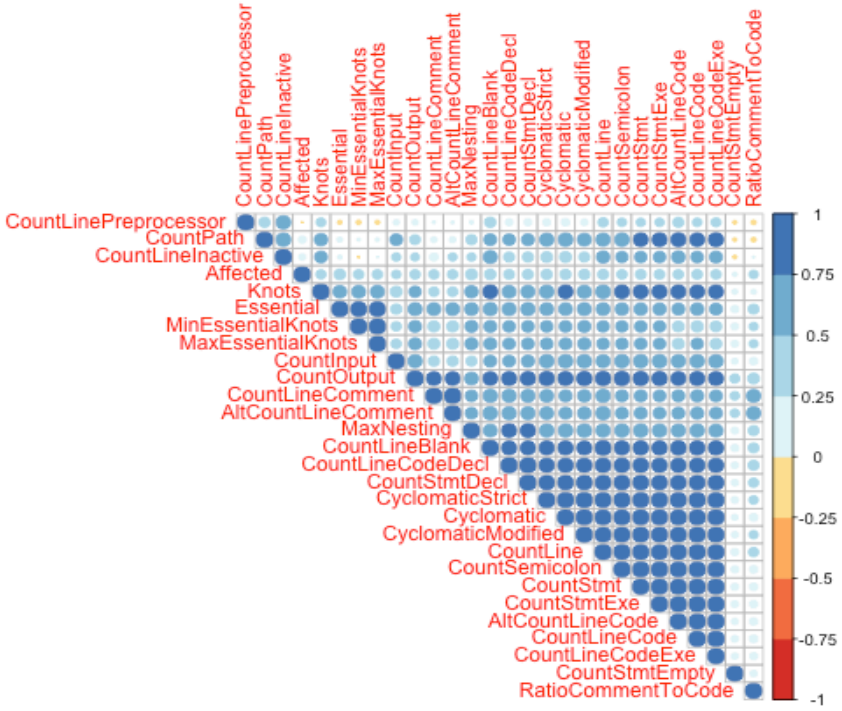
Conforme podemos ver, nem todas as métricas estão correlacionadas, mas destacam-se as métricas: Essential, Knots MaxNesting, CountStmtDecl, CountStmtExe, CountStmt, CountLineComment, AltCountLineComment, CountLineCodeDecl, CountLineCodeExe, CountLine, pois possuem uma boa quantidade de correlações, com no mínimo 4 outras métricas, sobressaindo ao número de métricas correlacionadas (com alta significância) em relação a base (glibc) que não está balanceada.

DATASET HTTPD NORMAL



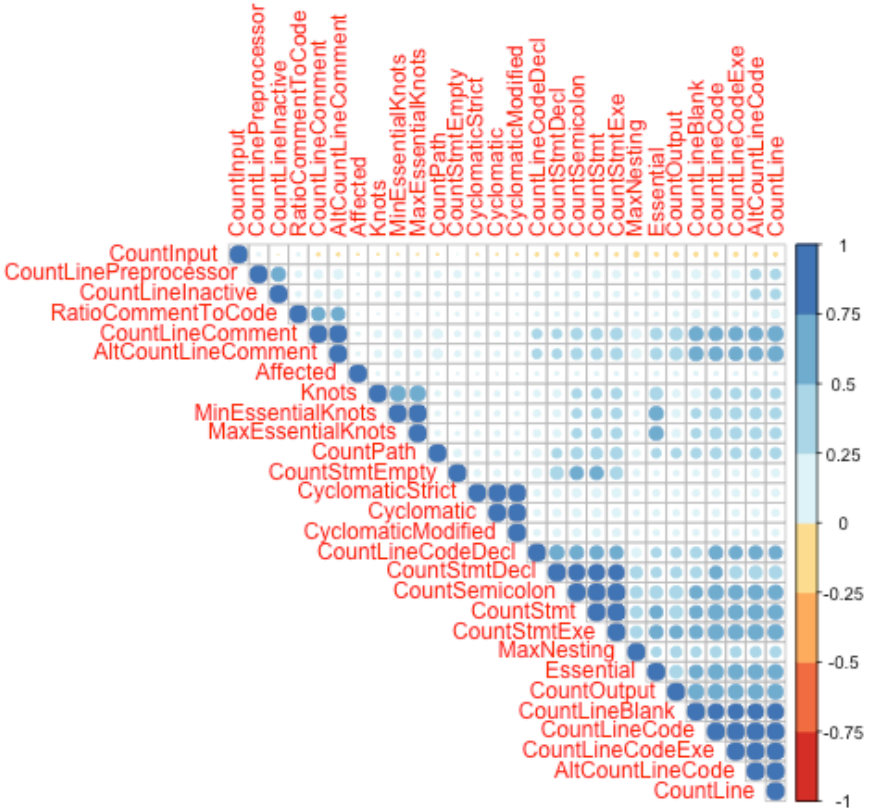
Na visualização acima pode-se ver que há um agrupamento entre as variáveis Essential e CountSemiColon que apresentam maior índice de significância nas correlações;

DATASET HTTPD BALANCEADO



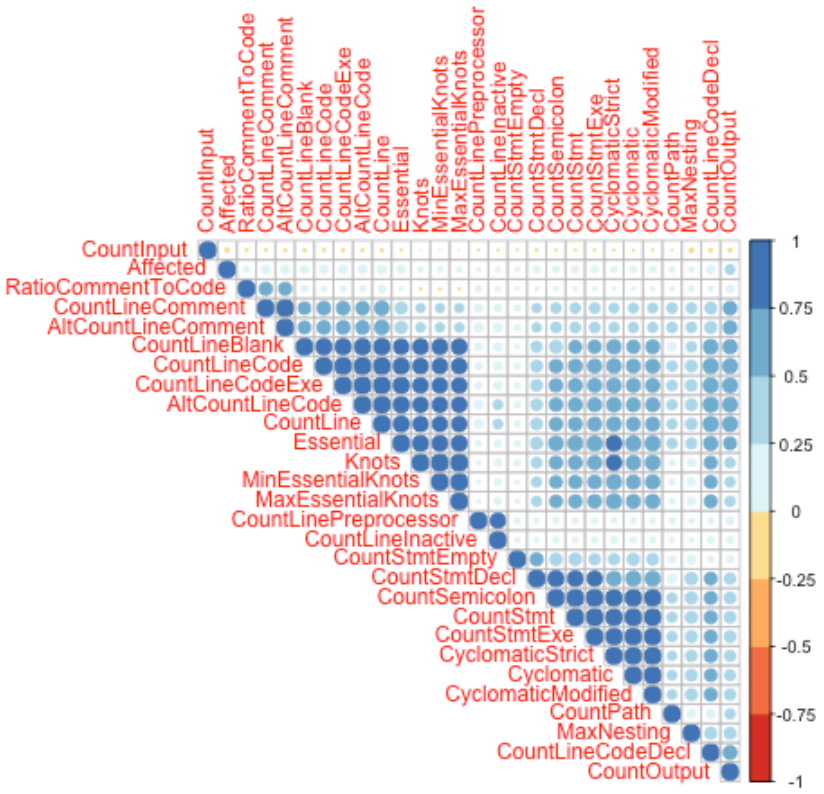
Comparando a base (httpd) balanceada X normal, pode-se ver que há um aumento no número de métricas correlacionadas, pois como exemplo temos a métrica CountSemiColon (número de ponto e virgula) que aumenta a medida que também ocorre o aumento das métricas CountStmt (número de instruções) e CountStmtExe (número de instruções executáveis), o que torna justificável.

DATASET KERNEL NORMAL



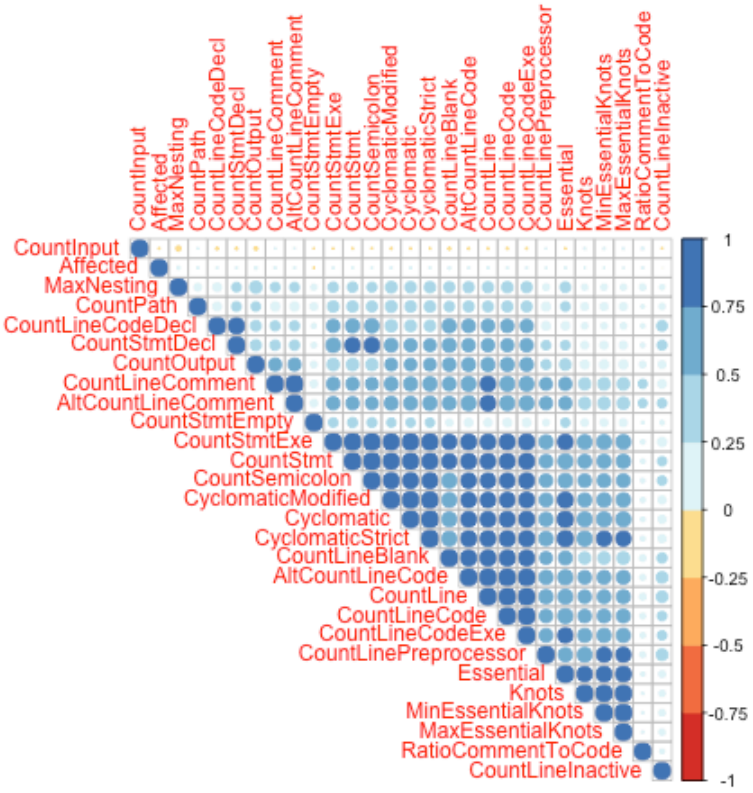
A visualização acima (que representa as correlações existentes entre as métricas no dataset não balanceado), indique pouca correlação entre as métricas, onde de fato analisado as correlações de maior relevância (CountLineBlank, CountLineCode, CountLineCodeExe) estão diretamente ligadas e contribuem proporcionalmente para o aumento da outra.

DATASET KERNEL BALANCEADO

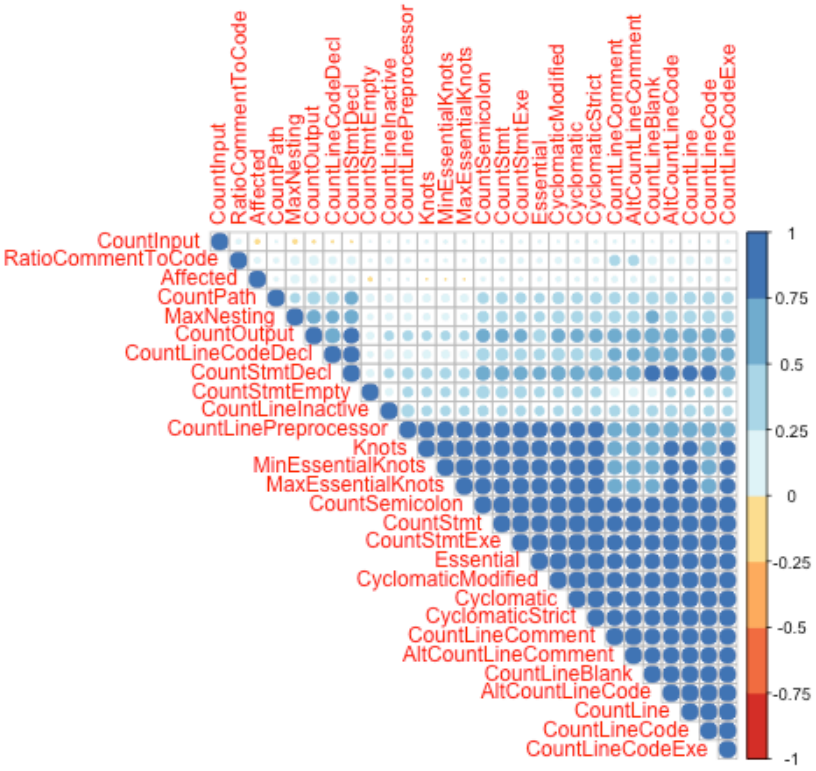


Comparando a base balanceada com a base normal, é percebido que as variáveis que foram vistas na correlações relevantes (CountLineBlank, CountLineCode, CountLineCodeExe) e (CountStmtDecl, CountSemiColon, CountStmt) se mantem correlacionadas na base de dados balanceada, gerando ainda o incremento de outras variáveis que se correlacionam.

DATASET MOZILA NORMAL

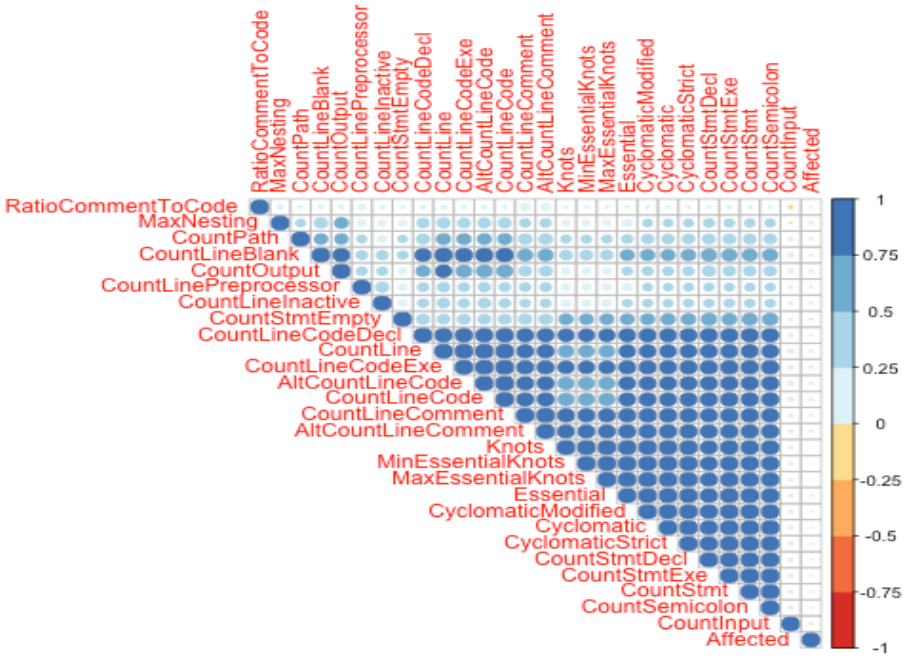


DATASET MOZILA BALANCEADO

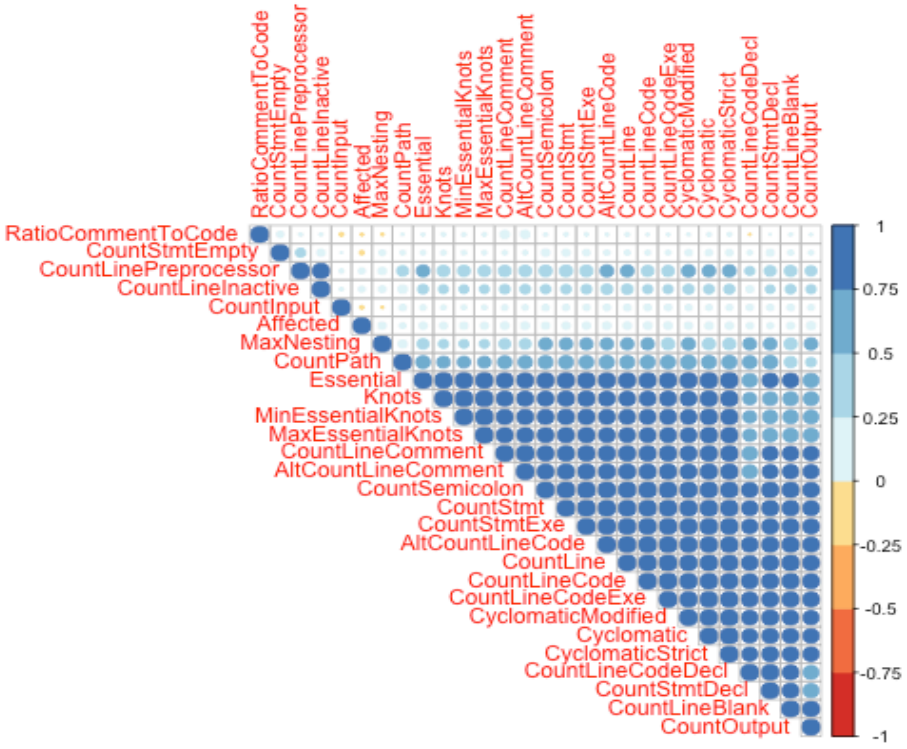


Observa-se uma maior incidência de métricas correlacionadas comparando a base balanceada com a base normal. Onde mais de 50% das métricas estão correlacionadas, e são de alta relevância.

DATASET XEN NORMAL



DATASET XEN BALANCEADO



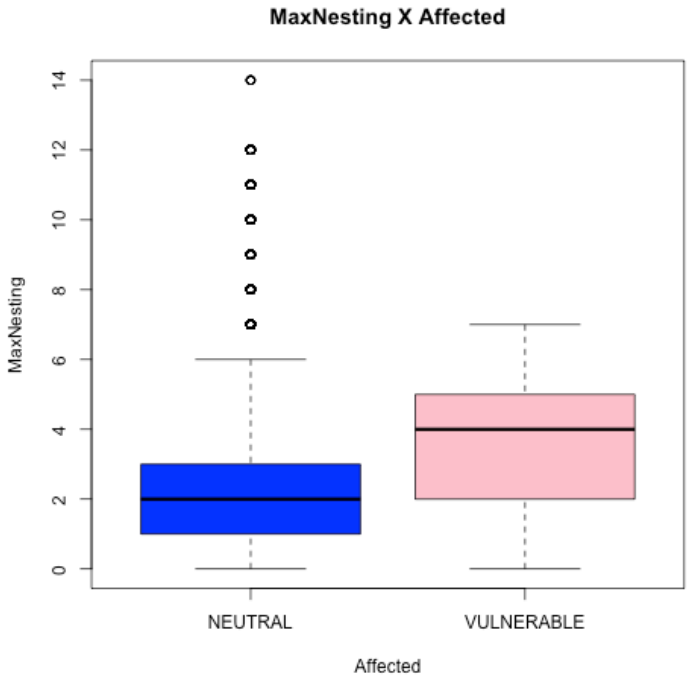
Na visualização referente as correlações do dataset xen, nota-se que há uma boa correlação entre as métricas, praticamente mantém-se as correlações (relevantes) em ambas as bases.

2. Are the software metrics able to represent functions with reported vulnerabilities?

(As métricas de software são capazes de representar funções com vulnerabilidades relatadas?)

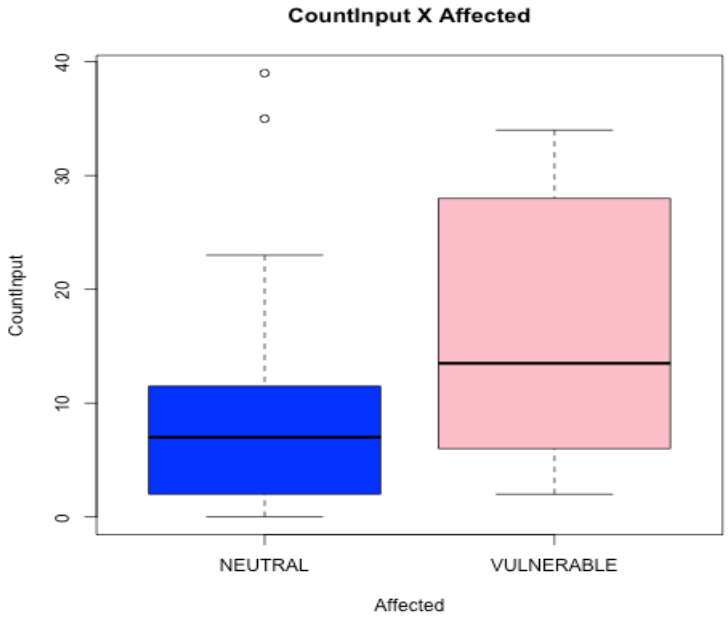
Para responder essa pergunta foram gerados gráficos bloxplots para analisar se as métricas possuem eficiência na identificação de vulnerabilidades, e como teste estatístico utilizamos o teste t-test, aplicando-o nas amostras (Neutral e Vulnerability).

DATASET GLIBC NORMAL



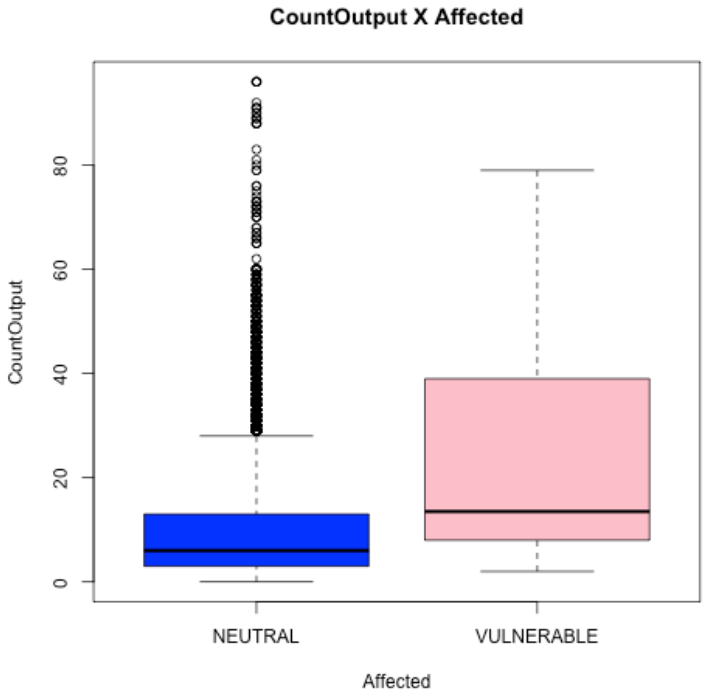
Observando o gráfico boxplot acima, podemos ver que a amostra VULNERABLE apresenta maior mediana, onde ocorre uma maior variabilidade nesses dados. Quanto a amostra NEUTRAL, vemos que existem alguns outliers, sendo eles os valores discrepantes no conjunto de dados.

DATASET GLIBC BALANCEADO



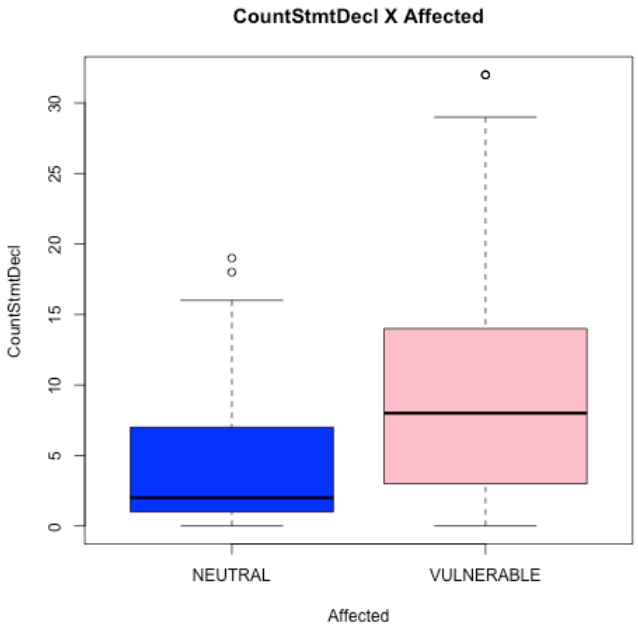
Quanto ao gráfico acima, vemos que a métrica CountInput, a amostra VULNERABLE apresenta quartis esparsos, contendo mais de 10 expressões de entrada, e a amostra NEUTRAL apresenta uma alta quantidade de outliers, o que impede diferenciar os perfis das amostras.

DATASET HTTPD NORMAL



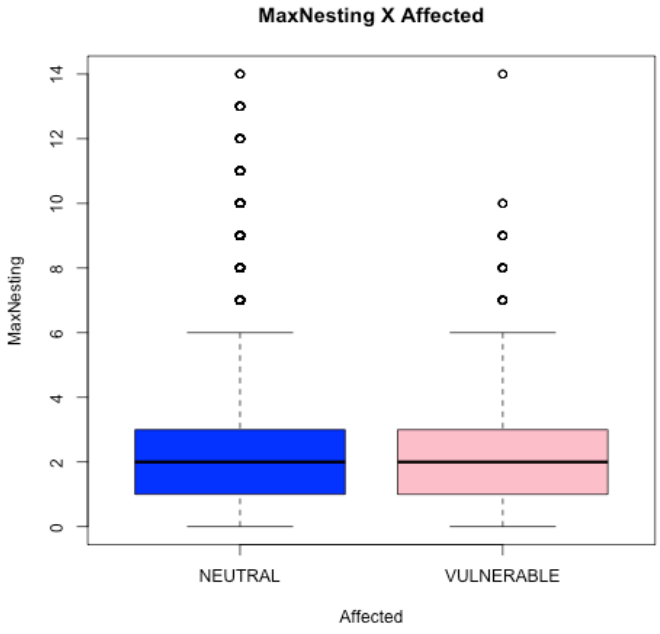
Na amostra VULNERABLE, podemos ver que a mediana maior, o que indica que metade das funções possuem vulnerabilidade. Quanto a amostra NEUTRAL, a presença de outliers não permite diferenciar o comportamento das amostras.

DATASET HTTPD BALANCEADO



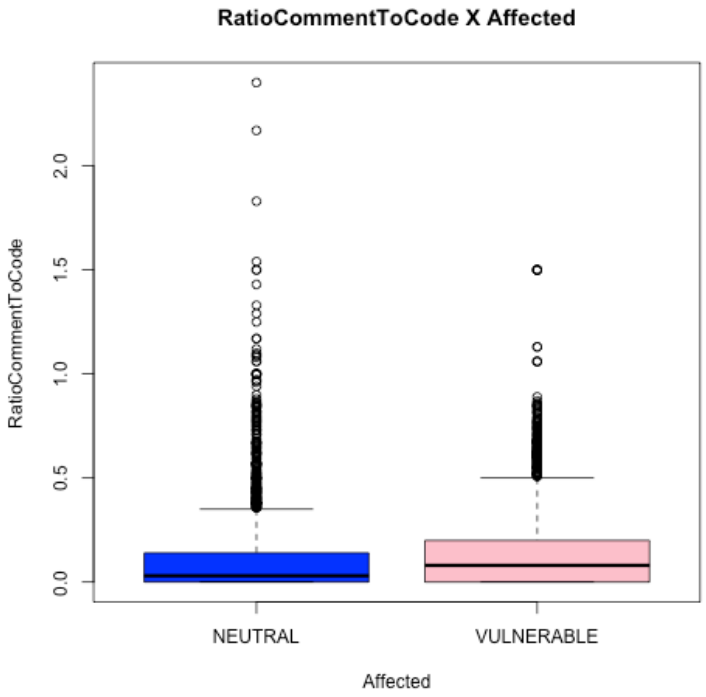
A representação acima (VULNERABLE) apresenta uma mediana maior, onde 50% das funções ficam expostas. E a amostra NEUTRAL traz consigo uma mediana baixa, onde a maior concentração de instruções encontram-se acima da mediana 50%.

DATASET KERNEL NORMAL

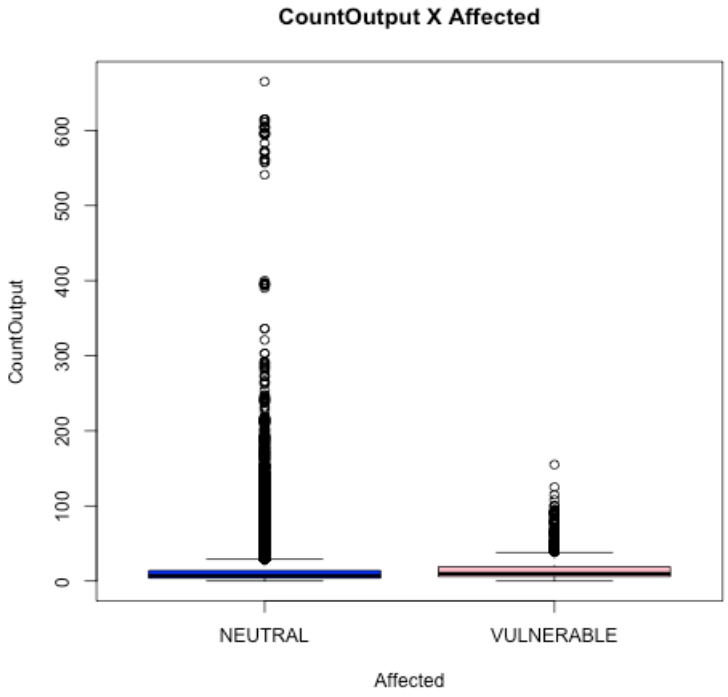


Na representação acima, devido a não apresentação de divergências, não se pode afirmar se há ou não vulnerabilidade nesse caso.

DATASET KERNEL BALANCEADO

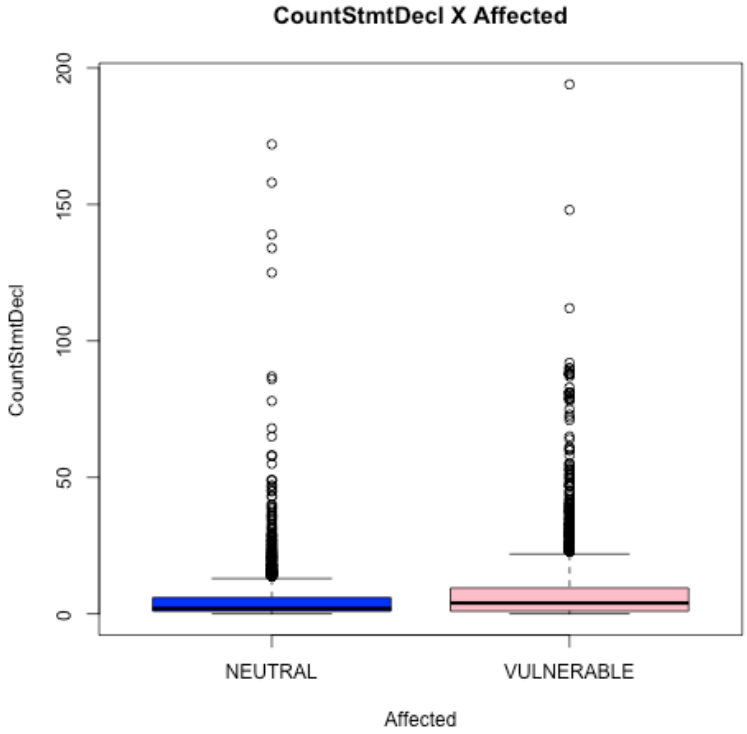


DATASET MOZILA NORMAL



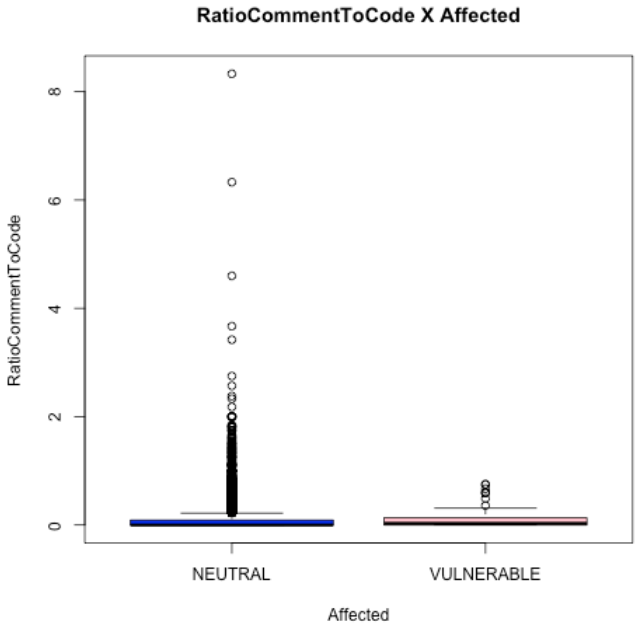
Na amostragem acima podemos ver uma semelhança entre as métricas, contudo devido a grande quantidade de outliers, fica difícil precisar se há ou não vulnerabilidade.

DATASET MOZILA BALANCEADO

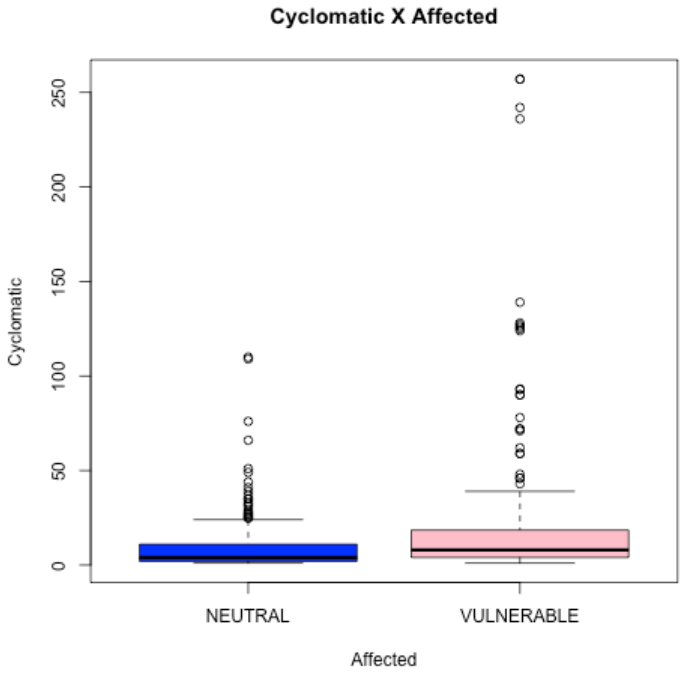


Na visualização acima podemos ver uma semelhança entre as métricas, apesar que a amostra VULNERABLE possuir uma mediana levemente acentuada, não é possível precisar se há ou não vulnerabilidade, devido a grande quantidade de outliers presente nas duas amostras.

DATASET XEN NORMAL



DATASET XEN BALANCEADO

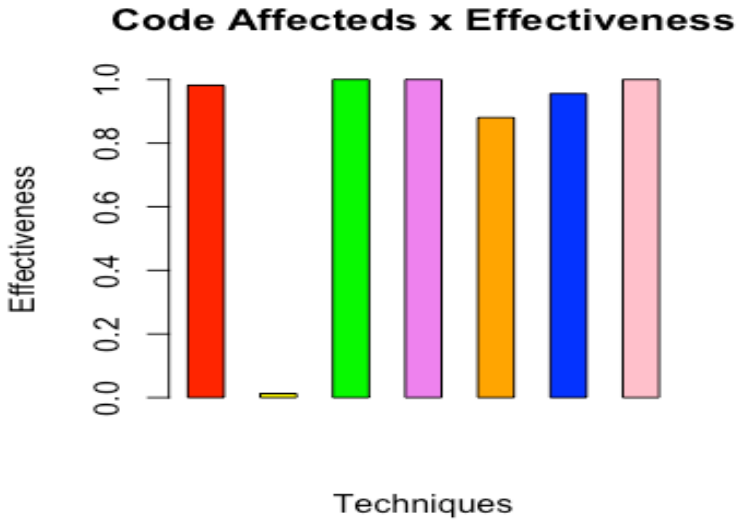


3. How effective are machine learning techniques to predict vulnerable functions?

(Quão eficazes são as técnicas de machine learning para prever funções vulneráveis?)

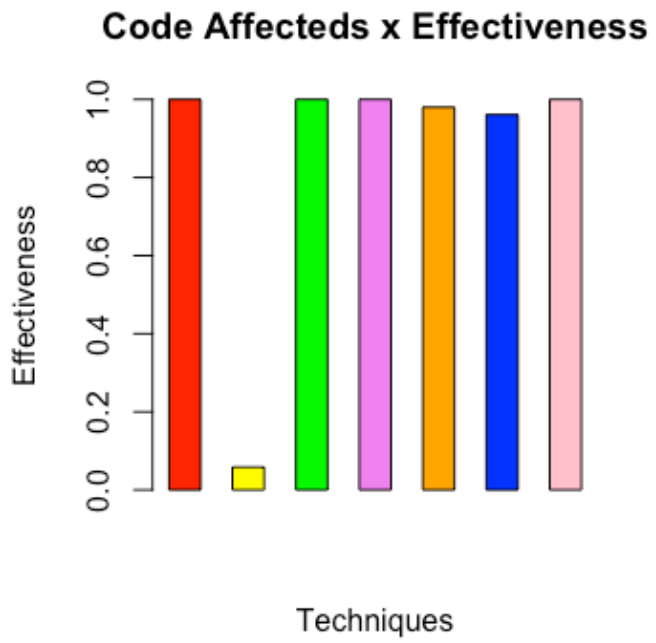
Cores	Algoritmo
Vermelho	J48
Amarelo	NaiveBayes
Verde	SVM
Violeta	OneR
Laranja	JRip
Blue	RandomForest
Rosa	SMO

DATA SET GLIBC BALANCEADO



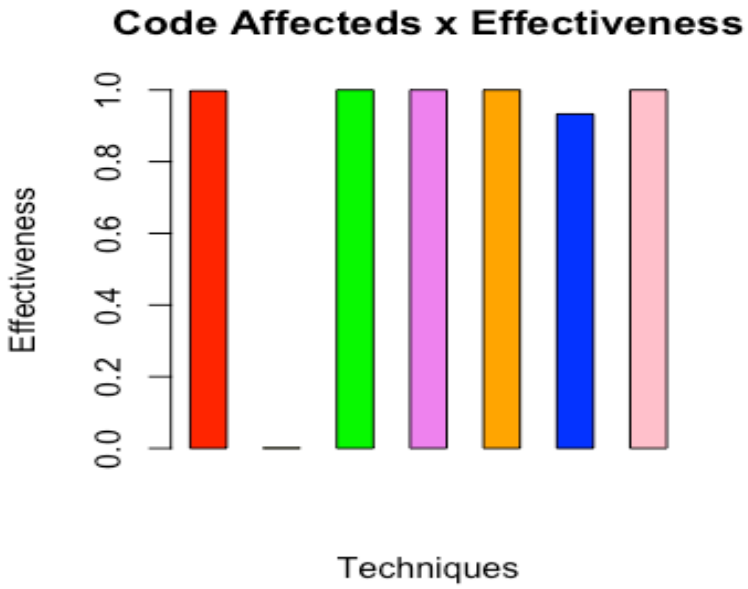
	Precision	Recall	F-measure
J48	1	0.966666667	0.98181818
NaiveBayes	1	0.002551916	0.01261401
SVM	1	1.0000000000	1.0000000000
oneR	1	1.0000000000	1.0000000000
JRip	1	0.796190476	0.88030303
RandomForest	1	0.920000000	0.95555556
SMO	1	1.0000000000	1.0000000000

DATA SET HTTPD BALANCEADO



	Precision	Recall	F-measure
J48	1	1.00000000	1.00000000
NaiveBayes	1	0.03010218	0.05830584
SVM	1	1.0000000000	1.0000000000
oneR	1	1.0000000000	1.0000000000
JRip	1	0.96181818	0.97994987
RandomForest	1	0.93000000	0.96090226
SMO	1	1.0000000000	1.0000000000

DATA SET XEN BALANCEADO



	Precision	Recall	F-measure
J48	1	0.9958333333	0.997894737
NaiveBayes	1	0.0002128891	0.001063634
SVM	1	1.0000000000	1.0000000000
oneR	1	1.0000000000	1.0000000000
JRip	1	1.0000000000	1.0000000000
RandomForest	1	0.8762867147	0.932971739
SMO	1	1.0000000000	1.0000000000

Observamos que de fato há uma pequena variação quanto a eficácia das técnicas para prever vulnerabilidades, na sua maioria o comportamento dessas técnicas não muda significativamente, fazendo com que haja uma bom aproveitamento quanto a analise. Contudo o que se mostro menos viável foi o NaiveBayes (amarelo), pois como podemos ver na sia representação gráfica, não se mostra tão significante quanto as outras.