



UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

INTELIGÊNCIA COMPUTACIONAL

---

## Breast Cancer Wisconsin

---

*Nomes:*

Aramys Almeida Matos

Luís Gustavo Oliveira Silva

*Professor:*

Alexandre Evsukoff

## 1 Introdução

Este trabalho foi desenvolvido sobre o *dataset Breast Cancer Wisconsin (Diagnostic)*, fornecido pelo *UCI Machine Learning*. As variáveis deste *dataset* são características computadas a partir de imagens digitalizadas de exames de mama por punção aspirativa por agulha fina. Os dados descrevem características do núcleo das células presentes na imagem.

## 2 Caracterização

### 2.1 Dados

Inicialmente é importante ressaltar que o conjunto de dados gerado pelos exames é tridimensional. Para cada registro (paciente) existe um conjunto de células, cada uma com os seguintes atributos.

1. Raio (média das distâncias do centro para pontos no perímetro)
2. Textura
3. Perímetro
4. Área
5. Suavidade
6. Compacidade
7. Concavidade
8. Pontos côncavos
9. Simetria
10. Dimensão Fractal

De forma a eliminar a terceira dimensão (do conjunto de células), para cada conjunto de células o *UCI Machine Learning* forneceu o dataset com valores de média, desvio(erro) padrão e o pior valor dos atributos no universo das células, resultando em 30 variáveis de entrada. O dataset conforme obtido possui as variáveis como segue:

- |                     |                            |                          |
|---------------------|----------------------------|--------------------------|
| 1. radius_mean      | 8. concave points_mean     | 15. smoothness_se        |
| 2. texture_mean     | 9. symmetry_mean           | 16. compactness_se       |
| 3. perimeter_mean   | 10. fractal_dimension_mean | 17. concavity_se         |
| 4. area_mean        | 11. radius_se              | 18. concave points_se    |
| 5. smoothness_mean  | 12. texture_se             | 19. symmetry_se          |
| 6. compactness_mean | 13. perimeter_se           | 20. fractal_dimension_se |
| 7. concavity_mean   | 14. area_se                | 21. radius_worst         |

22. texture_worst	25. smoothness_worst	28. concave points_worst
23. perimeter_worst	26. compactness_worst	29. symmetry_worst
24. area_worst	27. concavity_worst	30. fractal_dimension_worst

A variável de saída é o diagnóstico (maligno ou benigno codificados como  $-1$  e  $1$  respectivamente). Existe uma coluna com o ID do paciente que foi eliminada por ser irrelevante. O dataset apresenta 357 amostras benignas e 212 amostras malignas.

## 2.2 Estatísticas Básicas e Histogramas

- Radius

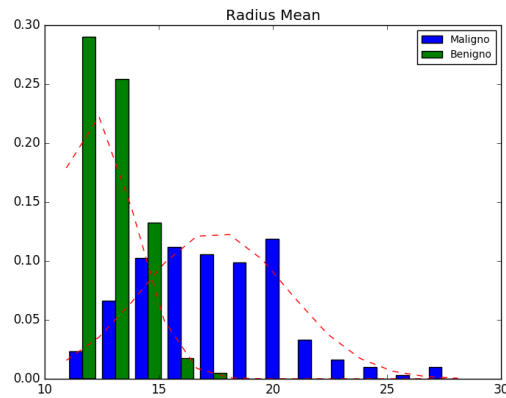


Figura 1: Mean

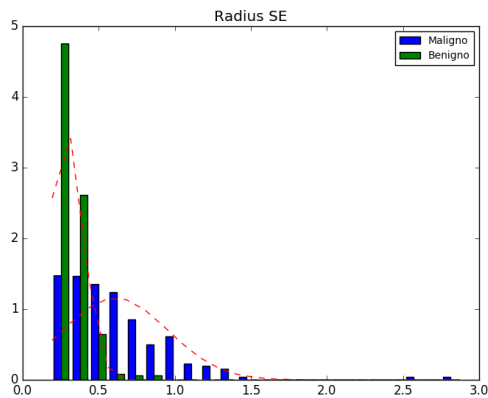


Figura 2: Standard Error

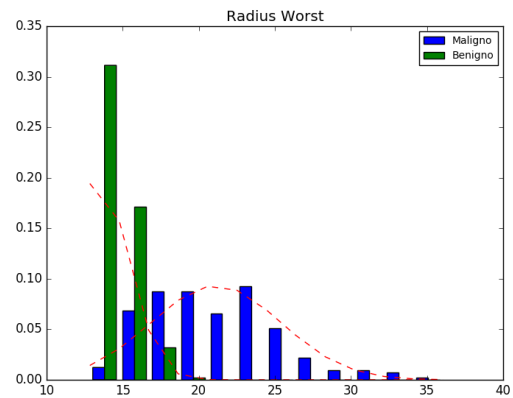


Figura 3: Worst

Tabela 1: Radius

	radius_mean	radius_se	radius_worst
Máximo	28.11	2.873	36.04
Mínimo	6.981	0.1115	7.93
Média	14.12729	0.405172	16.26919
Desvio padrão	3.524049	0.277313	4.833242
Percentil 25	11.7	0.2324	13.01
Percentil 50	13.37	0.3242	14.97
Percentil 75	15.78	0.4789	18.79

Análise: Para a variável Radius mean, vemos que a maioria de seus valores se concentram mais próximos da média que é 14,13. Para Radius Standard Error, também têm um comportamento semelhante a uma função de cauda longa, porém não temos a presença de valores no intervalo entre 1,6 e 2,4. Já para a variável Radius Worst, tem um comportamento semelhante à variável Radius mean.

- Texture

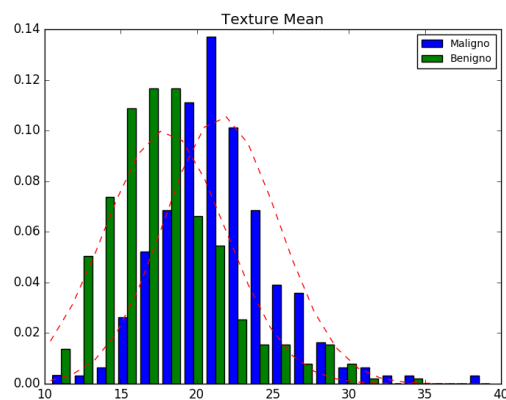


Figura 4: Mean

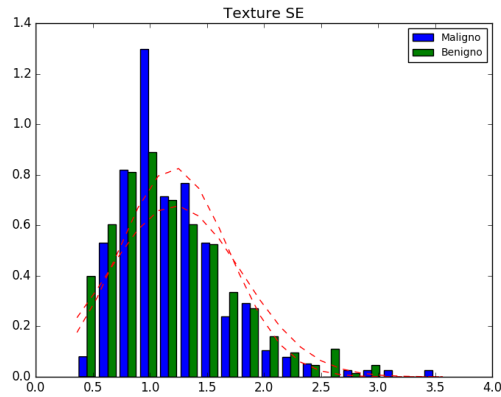


Figura 5: Standard Error

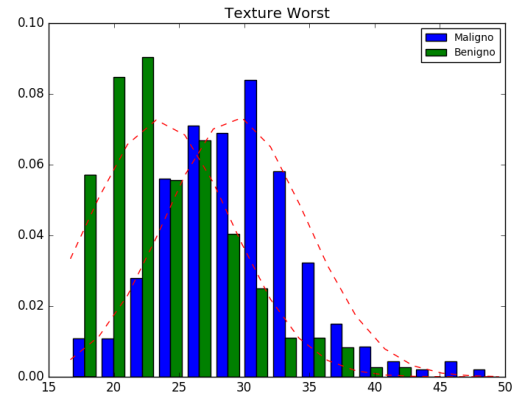


Figura 6: Worst

Tabela 2: Texture

	texture__mean	texture__se	texture__worst
Máximo	39.28	4.885	49.54
Mínimo	9.71	0.3602	12.02
Média	19.28964851	1.216853427	25.67722
Desvio padrão	4.301035768	0.551648393	6.146258
Percentil 25	16.17	0.8339	21.08
Percentil 50	18.84	1.108	25.41
Percentil 75	21.8	1.474	29.72

Análise: Podemos perceber que a variável Texture Mean, tem um comportamento que lembra a uma função Gaussiana, que de certa forma é espelhada em relação a média, com a exceção dos outliers. Em Texture Standart Error, a média é 1,22 e seus valores estão localizados próximos á média, porém temos uma certa quantidade de valores distantes, mesmo considerando o desvio padrão, e um valor máximo muito alto. Em Texture Worst, vemos que seu comportamento se assemelha a Texture Mean.

- Perimeter

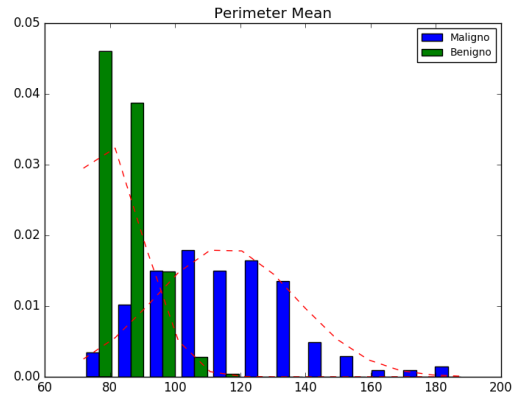


Figura 7: Mean

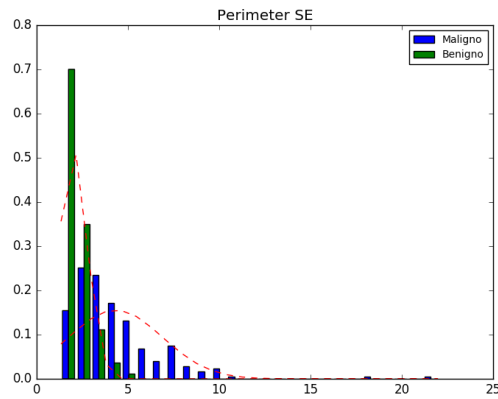


Figura 8: Standard Error

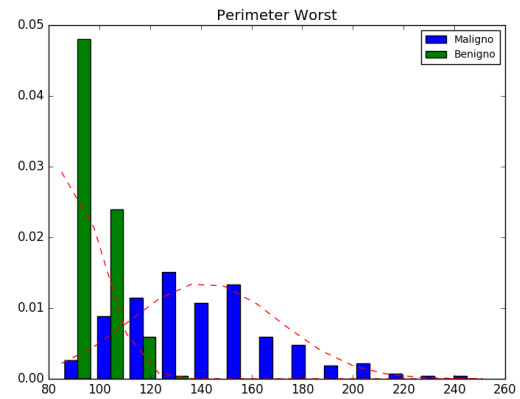


Figura 9: Worst

Tabela 3: Perimeter

	perimeter_mean	perimeter_se	perimeter_worst
Máximo	188.5	21.98	251.2
Mínimo	43.79	0.757	50.41
Média	91.96903339	2.866059227	107.2612
Desvio padrão	24.29898104	2.021854554	33.60254
Percentil 25	75.17	1.606	84.11
Percentil 50	86.24	2.287	97.66
Percentil 75	104.1	3.357	125.4

Análise: Em Perimeter Standard Error, vemos a presença de outliers, como por exemplo o valor máximo que é 21,98, enquanto sua média é 2.87. E em Perimeter Worst, vemos que possui um desvio padrão alto e seus valores estão distribuídos de forma distante da média.

- Area

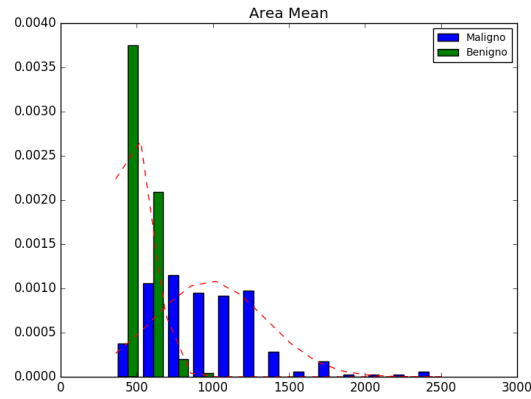


Figura 10: Mean

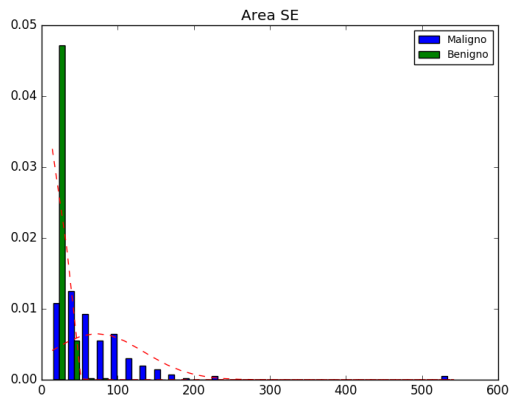


Figura 11: Standard Error

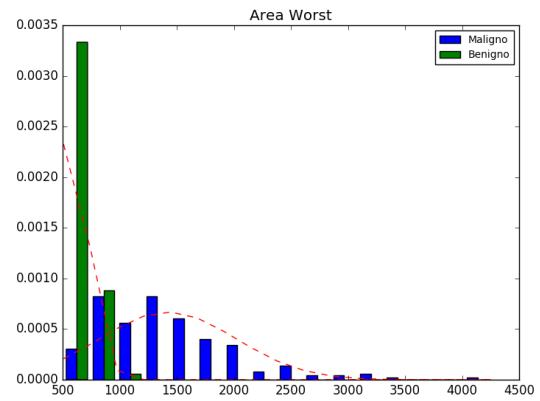


Figura 12: Worst

Tabela 4: Area

	area_mean	area_se	area_worst
Máximo	2501	542.2	4254
Mínimo	143.5	6.802	185.2
Média	654.8891037	40.33707909	880.5831283
Desvio padrão	351.9141292	45.49100552	569.3569927
Percentil 25	420.3	17.85	515.3
Percentil 50	551.1	24.53	686.5
Percentil 75	782.7	45.19	1084

Análise: Na variável Area Mean, vemos que ela possui um desvio padrão grande, sendo maior que a metade da média, assim como em Area Worst. Em Area Standard Error,

vemos que a variável tem um compartimento semelhante a uma função de cauda longa e temos um valor bem distante que é o valor máximo ( 2501,00).

- Smoothness

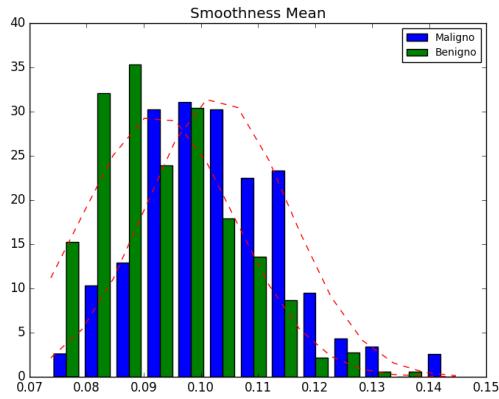


Figura 13: Mean

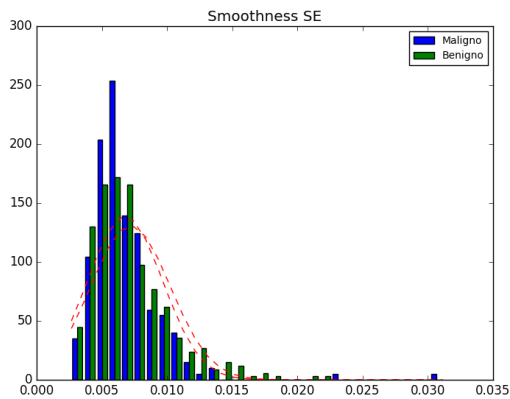


Figura 14: Standard Error

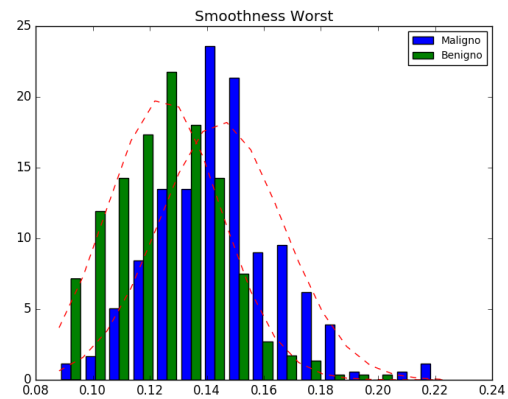


Figura 15: Worst

Tabela 5: Smoothness

	smoothness_mean	smoothness_se	smoothness_worst
Máximo	0.1634	0.03113	0.2226
Mínimo	0.05263	0.001713	0.07117
Média	0.096360281	0.007040979	0.132368594
Desvio padrão	0.014064128	0.003002518	0.022832429
Percentil 25	0.08637	0.005169	0.1166
Percentil 50	0.09587	0.00638	0.1313
Percentil 75	0.1053	0.008146	0.146



Análise: Podemos ver que tanto Smoothness Mean quanto em Worst, elas tem uma aparência semelhante a uma função Gaussiana e possuem um desvio padrão pequeno, já em Smoothness Standard Error, vemos que ela possui um desvio padrão alto e existe a presença de outliers como o seu valor máximo (0,16340).

- Compactness

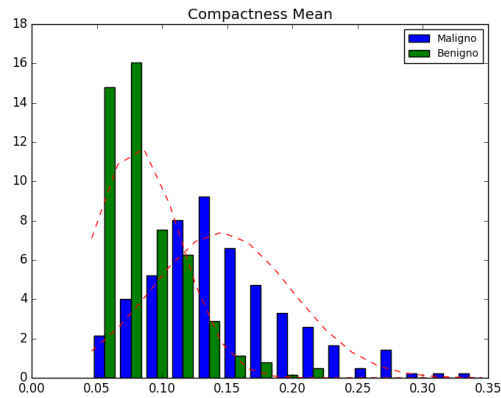


Figura 16: Mean

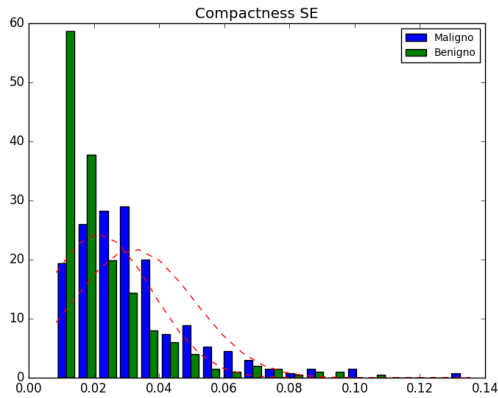


Figura 17: Standard Error

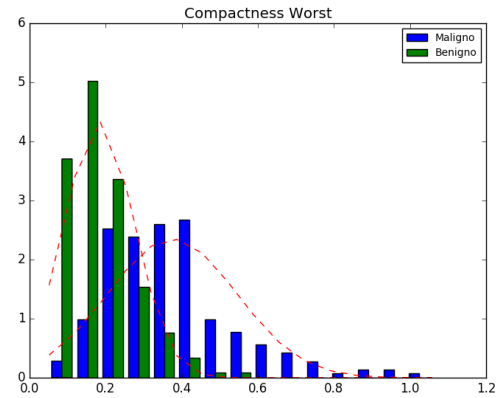


Figura 18: Worst

Tabela 6: Compactness

	compactness_mean	compactness_se	compactness_worst
Máximo	0.3454	0.1354	1.058
Mínimo	0.01938	0.002252	0.02729
Média	0.104340984	0.025478139	0.254265
Desvio padrão	0.052812758	0.017908179	0.157336
Percentil 25	0.06492	0.01308	0.1472
Percentil 50	0.09263	0.02045	0.2119
Percentil 75	0.1304	0.03245	0.3391

Análise: Aqui percebemos que as 3 variáveis possuem um desvio padrão alto e seus valores máximos se destoam bastante.

- Concavity

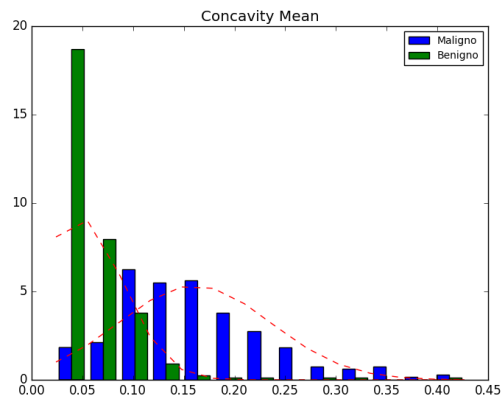


Figura 19: Mean

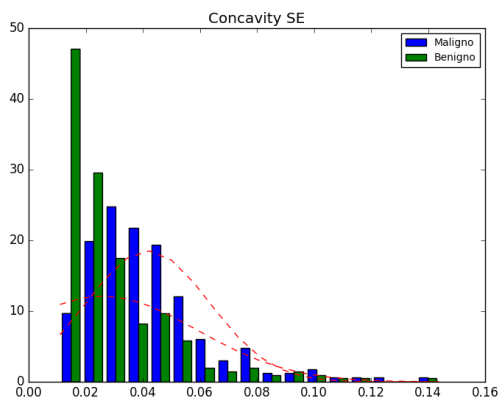


Figura 20: Standard Error

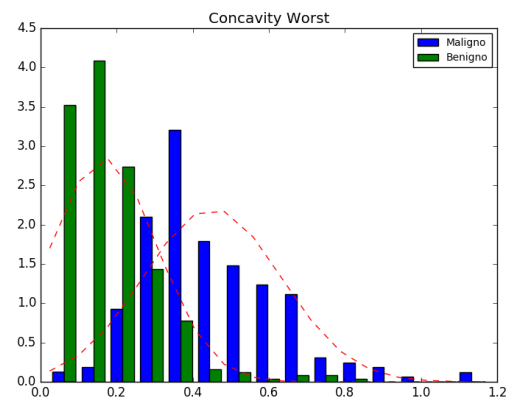


Figura 21: Worst

Tabela 7: Concavity

	concavity_mean	concavity_se	concavity_worst
Máximo	0.4268	0.396	1.252
Mínimo	0	0	0
Média	0.088799316	0.031893716	0.272188483
Desvio padrão	0.079719809	0.03018606	0.208624281
Percentil 25	0.02956	0.01509	0.1145
Percentil 50	0.06154	0.02589	0.2267
Percentil 75	0.1307	0.04205	0.3829

Análise: Nas 3 variáveis percebemos que o seus valores se concentram mais proximos de 0 e a ocorrência desses valores vão decaindo conforme se afastam de 0.

- Concave points

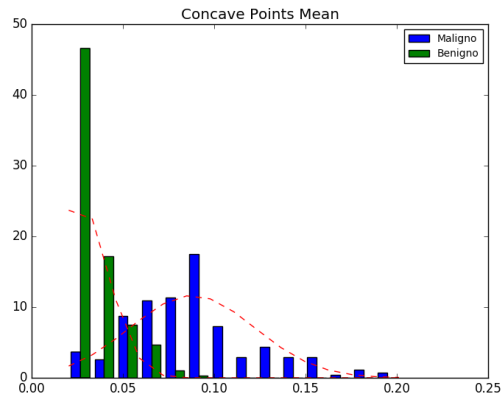


Figura 22: Mean

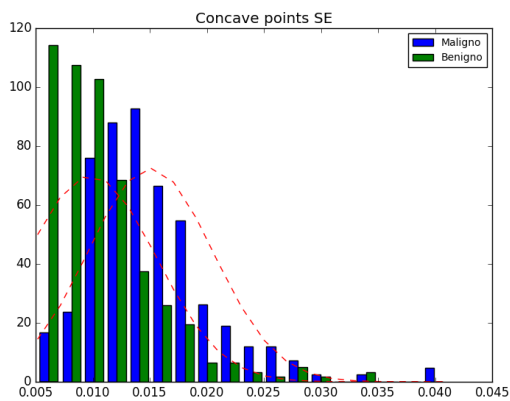


Figura 23: Standard Error

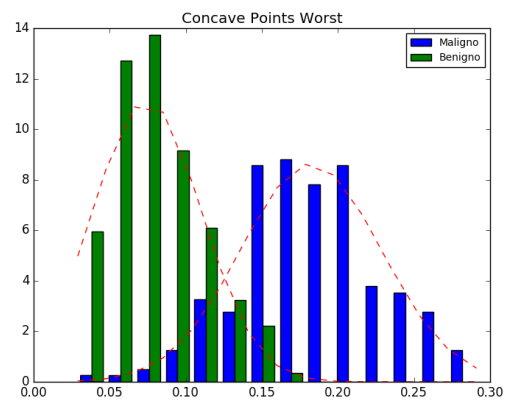


Figura 24: Worst

Tabela 8: Concave points

	concave points _mean	concave points _se	concave points _worst
Máximo	0.2012	0.05279	0.291
Mínimo	0	0	0
Média	0.048919146	0.011796	0.114606
Desvio padrão	0.038802845	0.00617	0.065732
Percentil 25	0.02031	0.007638	0.06493
Percentil 50	0.0335	0.01093	0.09993
Percentil 75	0.074	0.01471	0.1614

Análise: Aqui vemos que a variável Concave points mean, tem um comportamento semelhante à uma função de cauda longa e que a variável Concave Points Standard Error possui alguns outliers, como o valor máximo por exemplo.

- Symmetry

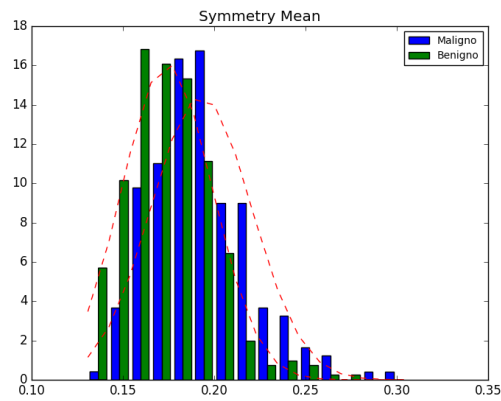


Figura 25: Mean

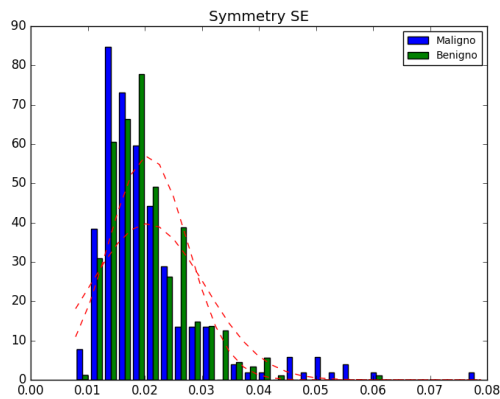


Figura 26: Standard Error

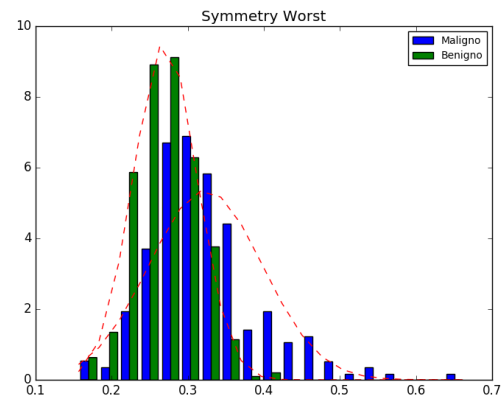


Figura 27: Worst

Tabela 9: Symmetry

	symmetry_mean	symmetry_se	symmetry_worst
Máximo	0.304	0.07895	0.6638
Mínimo	0.106	0.007882	0.1565
Média	0.181162	0.020542	0.290076
Desvio padrão	0.027414	0.008266	0.061867
Percentil 25	0.1619	0.01516	0.2504
Percentil 50	0.1792	0.01873	0.2822
Percentil 75	0.1957	0.02348	0.3179

Análise - A variável Symmetry mean possui um comportamento semelhante a uma função Gaussiana e tanto Symmetry Standard Error, quanto Worst possuem valores máximos distantes da média.

- Fractal Dimension

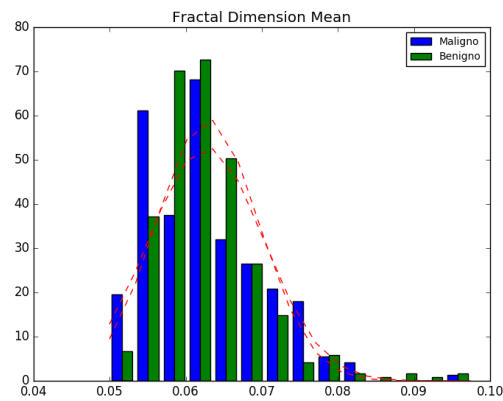


Figura 28: Mean

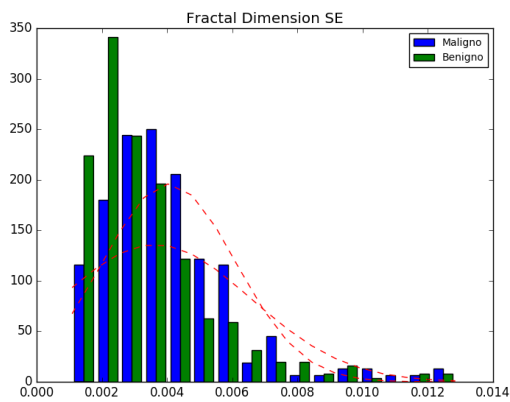


Figura 29: Standard Error

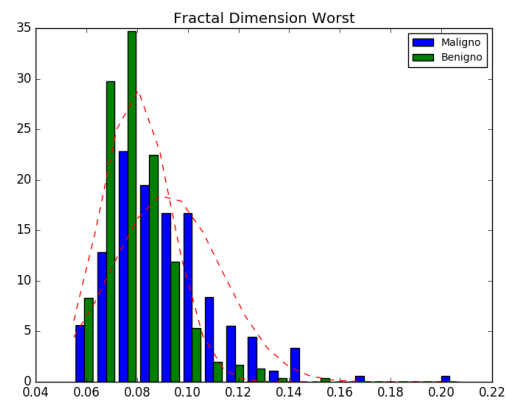


Figura 30: Worst

Tabela 10: Fractal dimension

	fractal_dimension_mean	fractal_dimension_se	fractal_dimension_worst
Máximo	0.09744	0.02984	0.2075
Mínimo	0.04996	0.000895	0.05504
Média	0.06279761	0.003795	0.083945817
Desvio padrão	0.007060363	0.002646	0.018061267
Percentil 25	0.0577	0.002248	0.07146
Percentil 50	0.06154	0.003187	0.08004
Percentil 75	0.06612	0.004558	0.09208

Análise: Podemos ver que apartir da média, a ocorrência dos valores das váreiaveis vão diminuindo conforme se distanciam da média.

A partir dos histogramas podemos avaliar que em geral não revelou características indesejáveis como distribuições multimodais. Conforme comentado algumas apresentaram assimetria.

### 2.3 Matriz de Correlação

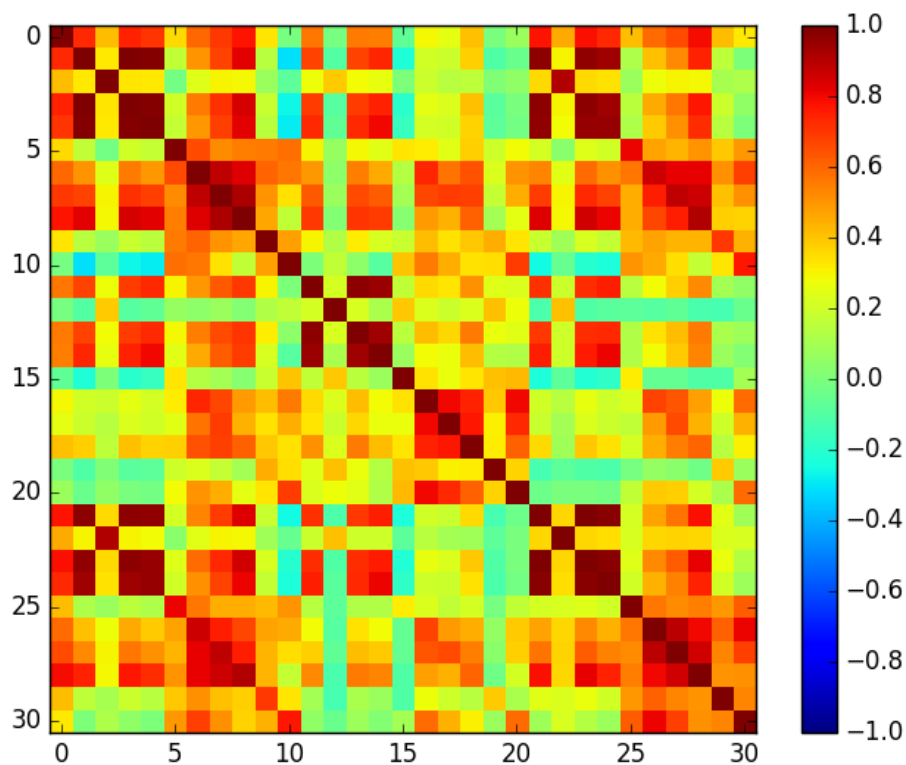


Figura 31: Matriz de Correlação

A partir da matriz podemos concluir que as seguintes variáveis estão fortemente correlacionadas (apresentam coeficiente de correlação acima de 0.9):

- Radius Mean, Perimeter Mean
- Radius Mean, Area Mean
- Radius Mean, Radius Worst
- Radius Mean, Perimeter Worst
- Radius Mean, Area Worst
- Texture Mean, Texture Worst
- Perimeter Mean, Area Mean
- Perimeter Mean, Radius Worst
- Perimeter Mean, Perimeter Worst
- Perimeter Mean, Area Worst
- Area Mean, Radius Worst
- Area Mean, Perimeter Worst
- Area Mean, Area Worst
- Concavity Mean, Concave Points Mean
- Concave Points Mean, Concave Points Worst
- Radius SE, Perimeter SE
- Radius SE, Area SE
- Perimeter SE, Area SE
- Radius Worst, Perimeter Worst
- Radius Worst, Area Worst
- Perimeter Worst, Area Worst

As seguintes variáveis apresentaram correlação negativa:

- Radius Mean, Fractal Dimension Mean
- Radius Mean, Texture SE
- Radius Mean, Smoothness SE
- Radius Mean, Symmetry SE
- Radius Mean, Fractal Dimension SE
- Texture Mean, Smoothness Mean
- Texture Mean, Fractal Dimension Mean
- Perimeter Mean, Fractal Dimension Mean
- Perimeter Mean, Texture SE
- Perimeter Mean, Smoothness SE
- Perimeter Mean, Symmetry SE
- Perimeter Mean, Fractal Dimension SE
- Area Mean, Fractal Dimension Mean
- Area Mean, Texture SE
- Area Mean, Smoothness SE
- Area Mean, Symmetry SE
- Area Mean, Fractal Dimension SE
- Fractal Dimension Mean, Area SE
- Fractal Dimension Mean, Radius Worst
- Fractal Dimension Mean, Texture Worst
- Fractal Dimension Mean, Perimeter Worst
- Fractal Dimension Mean, Area Worst
- Texture SE, Radius Worst
- Texture SE, Perimeter Worst
- Texture SE, Area Worst
- Texture SE, Smoothness Worst
- Texture SE, Compactness Worst
- Texture SE, Concavity Worst

- Texture SE, Concave Points Worst
- Texture SE, Symmetry Worst
- Texture SE, Fractal Dimension Worst
- Smoothness SE, Radius Worst
- Smoothness SE, Texture Worst
- Smoothness SE, Perimeter Worst
- Smoothness SE, Area Worst
- Smoothness SE, Compactness Worst
- Smoothness SE, Concavity Worst
- Smoothness SE, Concave Points Worst
- Smoothness SE, Symmetry Worst
- Symmetry SE, Radius Worst
- Symmetry SE, Texture Worst
- Symmetry SE, Perimeter Worst
- Symmetry SE, Area Worst
- Symmetry SE, Smoothness Worst
- Symmetry SE, Concave Points Worst
- Fractal Dimension SE, Radius Worst
- Fractal Dimension SE, Texture Worst
- Fractal Dimension SE, Perimeter Worst
- Fractal Dimension SE, Area Worst

## 2.4 Matriz de Distâncias

As figuras 32 e 33 mostram a matriz de distâncias, utilizando a Distância Euclidiana:

$$dist_E(\nu, v) = ||\nu - v||$$

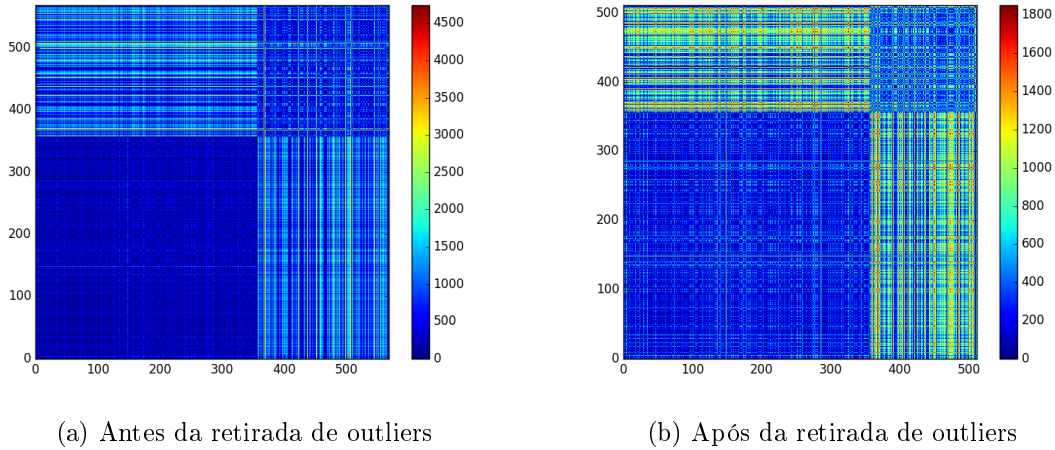


Figura 32: Matriz de distâncias

A matriz de distâncias para o conjunto de registros original pode ser observada na figura 32a. Foi realizado o processo de retirada de outliers baseado na distância média  $m_i = \frac{1}{N} \sum_{j=1}^N d_{ij}$ . Foram removidos os  $P_{out} = 10\%$  registros correspondentes aos maiores valores. A matriz de distâncias do conjunto de dados resultante pode ser vista na figura 32b.



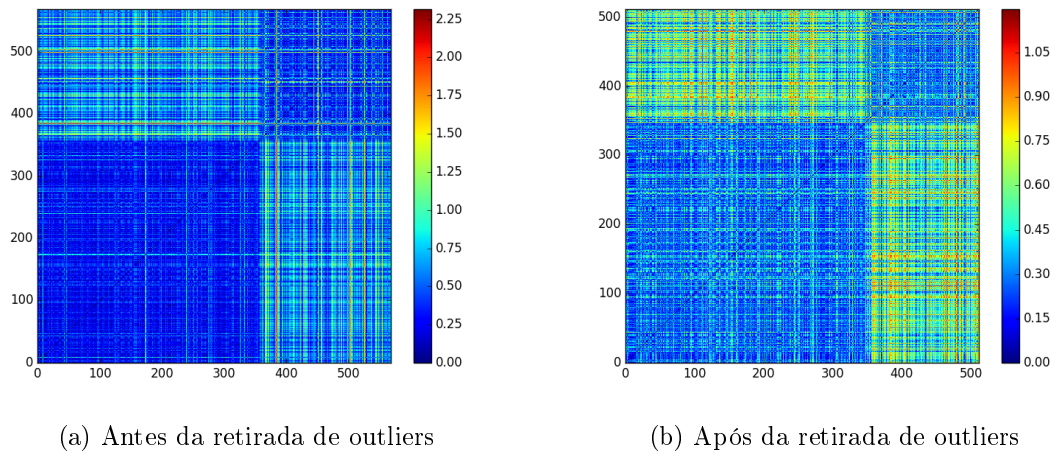


Figura 33: Matriz de distâncias com z-score

Como o conjunto de dados contém variáveis em unidades e escalas diferentes, o que dificulta a avaliação da matriz de distâncias pois os outliers de algumas variáveis acabam dominando. Assim, foi feita uma padronização das variáveis por meio da estimativa z-score e a matriz de distâncias resultante pode ser vista na figura 33.

### 3 Formulação do Problema

Consiste em um problema de classificação. Deseja-se desenvolver um classificador capaz de prever a classe do registro (maligno ou benigno).

Para solução deste problema foram aplicados os modelos de classificação a seguir e avaliados os resultados.

- Classificador Bayesiano Simples
- Classificador Bayesiano Quadrático
- Regressão Logística
- Perceptron
- Perceptron Múltiplas Camadas
- SVM

### 4 Metodologia

Para os testes será utilizada a metodologia de validação cruzada de 10 ciclos. A cada iteração o dataset é dividido em subconjuntos complementares. O modelo é ajustado com o conjunto de treinamento e aplicado para prever a saída do conjunto de teste. O resultado final é obtido como a média de cada iteração.

Para avaliar o desempenho do classificador, os resultados preditos pelo teste são comparado com os resultados reais e contabilizada a matriz de confusão. A partir dela são calculadas as seguintes métricas:

- Acurácia
- Erro global
- Precisão
- Recuperação

Métodos como Perceptron, Perceptron Múltiplas Camadas e SVM apresentaram grande sensibilidade a padronização das variáveis de entrada, no entanto, os demais métodos apresentaram resultados melhores sem padronização. Portanto, a título de comparação ambos os resultados são mostrados. No caso do MLP e SVM somente foi exibido o resultado com padronização pois caso contrário apresentavam problemas de convergência.

## 5 Apresentação da Tecnologia

Para a implementação e execução dos algoritmos foram usadas as seguintes bibliotecas da linguagem Python:

- *SciPy*: Ecossistema de softwares para matemática, ciência e engenharia. Contém os pacotes:
  - *NumPy*
  - *matplotlib*
  - *pandas*: Python Data Analysis Library
- *scikit-learn*: Machine Learning in Python

## 6 Classificador Bayesiano Simples

### 6.1 Formulação matemática

O Classificador Bayesiano baseia-se na aplicação do Teorema de Bayes com a suposição de independência entre cada par de variáveis, ou seja, que não há correlação entre as variáveis.

$$P(y|x_1 \dots x_n) = \frac{P(y)P(x_1, \dots, x_n|y)}{P(x_1, \dots, x_n)}$$

Onde:

$y$  é a variável de saída que identifica a classe

$x = [x_1, \dots, x_n]$  é o vetor de entrada

A decisão é feita pelo valor máximo da probabilidade a posteriori, dividindo o espaço em:

$$R_1 = x \in R, P(C_1|x) \geq P(C_2|x)$$

$$R_2 = x \in R, P(C_2|x) > P(C_1|x)$$

## 6.2 Resultados (sem padronização)

O dataset não é linearmente separável portanto os resultados não foram tão satisfatórios, já que o classificador Bayesiano Simples gera uma fronteira de decisão linear. Além disso, assume-se que a distribuição de probabilidade das características é Gaussiana, o que não se verifica para todas as características do dataset, como visto nos histogramas.

Tabela 11: Matriz de confusão

	$\hat{C}_1$ (Predita)	$\hat{C}_2$ (Predita)
$C_1$	18.80	2.40
$C_2$	1.20	34.50

- $ACC = 93.67\%$
- $ERR = 6.33\%$
- $AUC = 0.9266$
- $PRE(C1) = 0.94$
- $REC(C1) = 0.8868$
- $PRE(C2) = 0.9350$
- $REC(C2) = 0.9664$

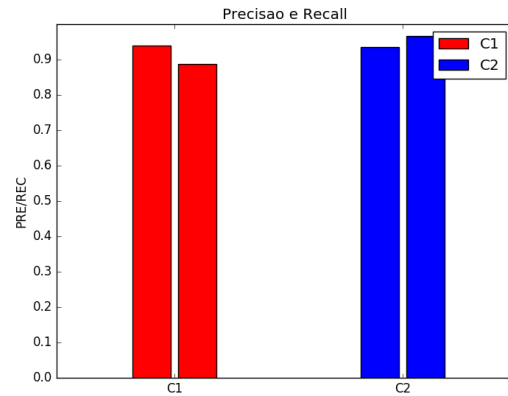


Figura 34: Precisão e Recall

## 6.3 Resultados (com padronização)

Tabela 12: Matriz de confusão

	$\hat{C}_1$ (Predita)	$\hat{C}_2$ (Predita)
$C_1$	18.9	2.3
$C_2$	1.9	33.8

- $ACC = 92.62\%$
- $ERR = 7.38\%$
- $AUC = 0.9191$
- $PRE(C1) = 0.9087$
- $REC(C1) = 0.8915$
- $PRE(C2) = 0.9363$
- $REC(C2) = 0.9468$

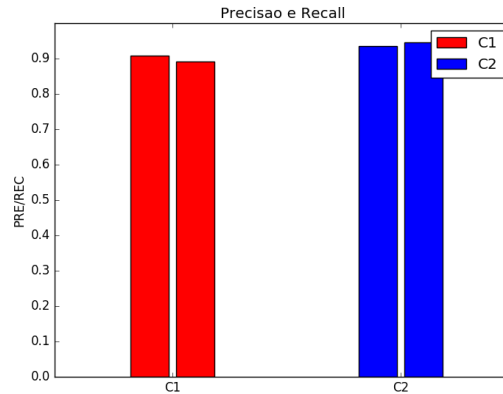


Figura 35: Precisão e Recall

## 7 Classificador Bayesiano Quadrático

### 7.1 Formulação Matemática

É um classificador com uma fronteira de decisão quadrática, gerado pela densidades condicionais dos dados e utilizando a regra de Bayes.

A decisão é calculada pela função discriminante:

$$g_i(x(t)) = \ln P(C_i|x(t)) = \ln p(x(t)|C_i) + \ln P(C_i)$$

Substituindo a expressão da distribuição normal multivariada, observa-se que a expressão da função discriminante é dominada por um termo quadrático.

### 7.2 Resultados

Verifica-se que os resultados foram melhores que os obtidos com Naive Bayes pois a superfície de decisão agora é quadrática, permitindo a classificação de modelos mais complexos. Neste caso os resultados com e sem padronização foram os mesmos.

Tabela 13: Matriz de confusão

	$\hat{C}_1$ (Predita)	$\hat{C}_2$ (Predita)
$C_1$	20.00	1.20
$C_2$	1.20	34.50

- $ACC = 95.78\%$
- $ERR = 4.22\%$
- $AUC = 0.9549$
- $PRE(C1) = 0.9434$
- $REC(C1) = 0.9434$
- $PRE(C2) = 0.9664$
- $REC(C2) = 0.9664$

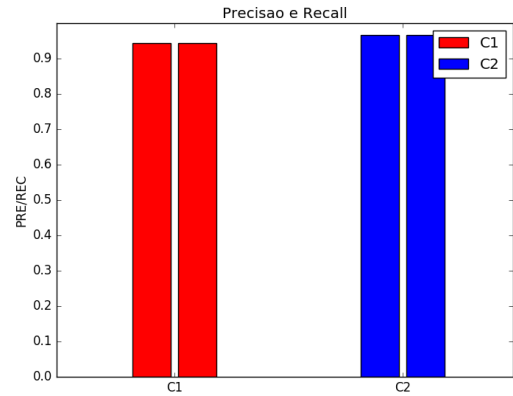


Figura 36: Precisão e Recall

## 8 Regressão Logística

### 8.1 Formulação Matemática

Na regressão logística, a saída do modelo é uma aproximação da probabilidade a posteriori.

A função discriminante é calculada pela função sigmoide, ou função logística ou logit:

$$g_i(x(t)|\theta_i) = \frac{1}{1 + \exp(\hat{x}(t)\theta_i)}$$

Onde:

$$\hat{x}(t) = [1, x(t)] \quad e \quad \theta_i = [\theta_{i0}, \theta_{i1}]^T$$

### 8.2 Resultados (sem padronização)

Tabela 14: Matriz de confusão

	$\hat{C}_1$ (Predita)	$\hat{C}_2$ (Predita)
$C_1$	20.00	1.20
$C_2$	1.10	34.60

- $ACC = 95.96\%$
- $ERR = 4.04\%$
- $AUC = 0.9563$
- $PRE(C1) = 0.9479$
- $REC(C1) = 0.9434$
- $PRE(C2) = 0.9665$
- $REC(C2) = 0.9692$

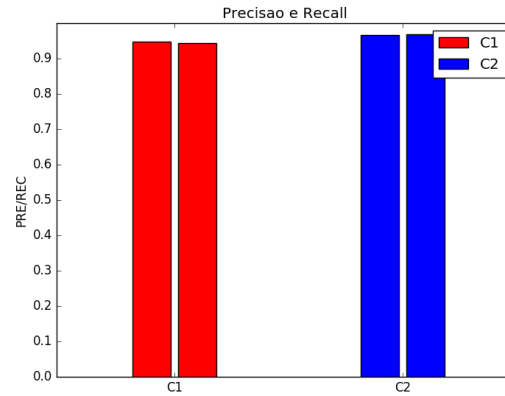


Figura 37: Precisão e Recall

### 8.3 Resultados (com padronização)

Tabela 15: Matriz de confusão

	$\hat{C}_1$ (Predita)	$\hat{C}_2$ (Predita)
$C_1$	19.9	1.3
$C_2$	1.6	34.1

- $ACC = 94.9\%$
- $ERR = 5.1\%$
- $AUC = 0.9469$
- $PRE(C1) = 0.9256$
- $REC(C1) = 0.9387$
- $PRE(C2) = 0.9633$
- $REC(C2) = 0.9552$

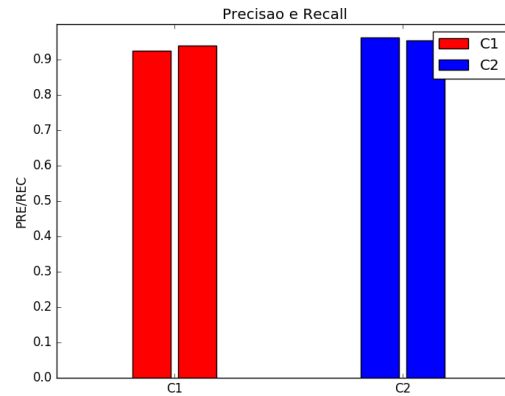


Figura 38: Precisão e Recall

## 9 Perceptron

### 9.1 Formulação Matemática

O Perceptron utiliza o modelo McCulloch-Pitts para o neurônio artificial. O processamento de cada unidade é dado por:

$$u(t) = h(z(t)) = h\left(\theta_0 + \sum_{i=1}^n x_i(t)\theta_i\right)$$

onde:

$u(t)$ : valor de ativação

$z(t)$ : potencial de ativações

$h$ : função de ativação

$x_i(t)$ : entradas do neurônio

A função custo do Perceptron é linear como mostrada na figura ??, ou  $Hinge(0)$ .

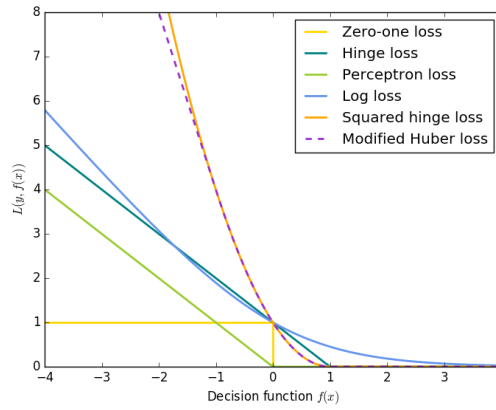


Figura 39: Funções Custo

### 9.2 Resultados (sem padronização)

Fica evidente que o Perceptron é bastante sensível a escala das variáveis, apresentando desempenho bastante inferior ao que apresentou com padronização.

Tabela 16: Matriz de confusão

	$\hat{C}_1$ (Predita)	$\hat{C}_2$ (Predita)
$C_1$	16.8	4.4
$C_2$	1.1	34.6

- $ACC = 90.33\%$
- $ERR = 9.67\%$
- $AUC = 0.8808$
- $PRE(C1) = 0.9385$
- $REC(C1) = 0.7925$
- $PRE(C2) = 0.8872$
- $REC(C2) = 0.9692$

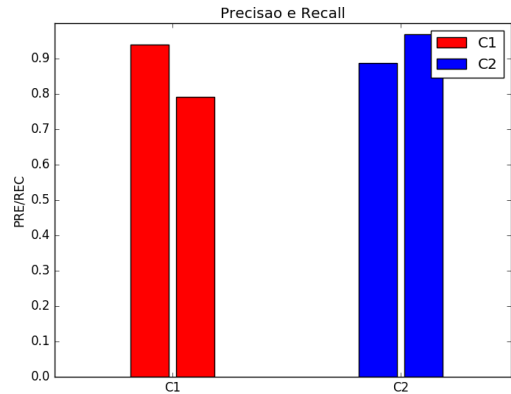


Figura 40: Precisão e Recall

### 9.3 Resultados (com padronização)

Tabela 17: Matriz de confusão

	$\hat{C}_1$ (Predita)	$\hat{C}_2$ (Predita)
$C_1$	20.40	0.80
$C_2$	1.40	34.30

- $ACC = 96.13\%$
- $ERR = 3.87\%$
- $AUC = 0.9615$
- $PRE(C1) = 0.9358$
- $REC(C1) = 0.9623$
- $PRE(C2) = 0.9772$
- $REC(C2) = 0.9608$

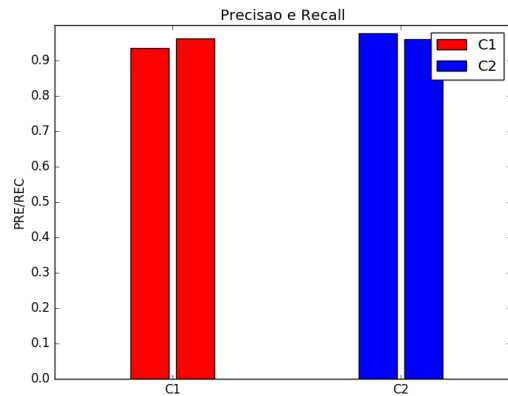


Figura 41: Precisão e Recall

## 10 Perceptron de Múltiplas Camadas (MLP)

### 10.1 Formulação Matemática

Uma rede-neural MLP apresenta uma camada de entrada que não realiza processamento com a dimensão do vetor de entrada, uma ou mais camadas intermediárias que realizam processamento e uma camada de saída que, num problema de classificação é o vetor com as estimativas



das variáveis indicadoras. O modelo McCulloch-Pitts é utilizado nas unidades das camadas intermediárias e de saída. Assim, o MLP é capaz de aprender uma função  $f(\cdot) : R^m \rightarrow R^o$ , onde  $m$  é dimensão do vetor de entrada e  $o$  é a dimensão do vetor de saída. O Perceptron de Múltiplas Camadas é capaz de aprender modelos não lineares.

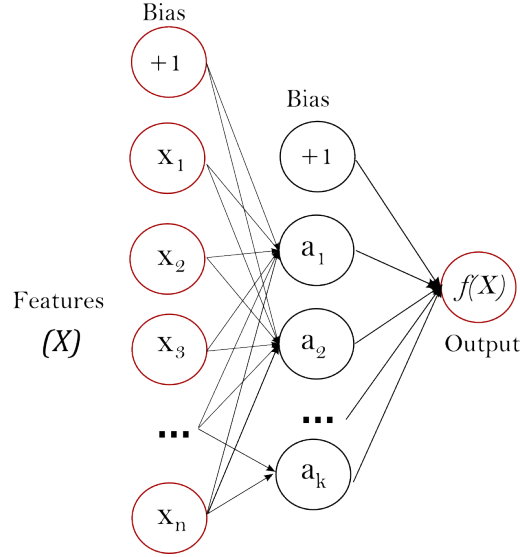


Figura 42: Rede neural MLP

A função custo utilizada é a Entropia cruzada, que para cada elemento é dada por:

$$l(\hat{y}, y, \theta) = -y \ln \hat{y} - (1 - y) \ln (1 - \hat{y}) + \alpha \|\theta\|_2^2$$

Onde  $\alpha \|\theta\|_2^2$  é um termo de regularização,  $\alpha > 0$  um parametro que controla a magnitude da penalidade e  $\theta$  é o vetor de parâmetros.

A função de ativação escolhida para os neurônios das camadas intermediárias foi a Tangente hiperbólica:

$$u(t) = \frac{1 - \exp(-z(t))}{1 + \exp(-z(t))}$$

Método de ajuste dos parâmetros escolhido foi o Gradiente descendente estocástico:

$$\theta^{i+1} = \theta^i - \eta \nabla l(\theta)^i$$

Onde  $i$  é o número da iteração e  $\eta$  a taxa de aprendizado.

Depois de calculada função custo, o ajuste é propagado para as camadas anteriores através do Backpropagation.

#### Observações:

Durante a execução do algoritmo notou-se que os resultados eram diferentes a cada execução. Isso se deve a inicialização aleatória dos parametros. Quando existem camadas intermediárias a função custo não é convexa, portanto existe mais de um mínimo local. Isso pode ser resolvido configurando manualmente a semente do gerador aleatório.

Além disso, o método é bastante sensível a escala das variáveis de entrada, tornando necessária, mais que em outros métodos, a padronização das variáveis pois caso contrário o

método apresenta problemas de convergência. Foi feita uma padronização de modo a obter média zero e desvio padrão 1. A padronização é calculada no conjunto de treinamento e a mesma transformação é aplicada para o conjunto de teste.

## 10.2 Resultados - (21)

Para este teste foi utilizada uma heurística para obter o número de camadas intermediárias:

$$N = \frac{2}{3}(m + o)$$

Sendo  $m$  a dimensão da entrada e  $o$  a dimensão da saída.

Tabela 18: Matriz de confusão

	$\hat{C}_1$ (Predita)	$\hat{C}_2$ (Predita)
$C_1$	20.20	1.00
$C_2$	0.40	35.30

- $ACC = 97.35\%$
- $ERR = 2.65$
- $AUC = 0.9708\%$
- $PRE(C1) = 0.9806$
- $REC(C1) = 0.9528$
- $PRE(C2) = 0.9725$
- $REC(C2) = 0.9888$

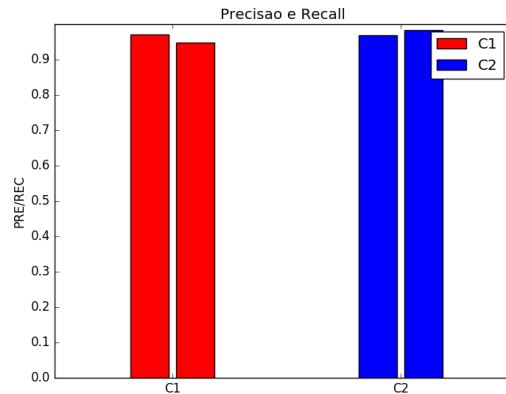


Figura 43: Precisão e Recall

## 10.3 Resultados - (100)

Tabela 19: Matriz de confusão

	$\hat{C}_1$ (Predita)	$\hat{C}_2$ (Predita)
$C_1$	20.40	0.80
$C_2$	0.30	35.40

- $ACC = 98.07\%$
- $ERR = 1.93\%$
- $AUC = 0.9769$
- $PRE(C1) = 0.9855$
- $REC(C1) = 0.9623$
- $PRE(C2) = 0.9779$
- $REC(C2) = 0.9916$

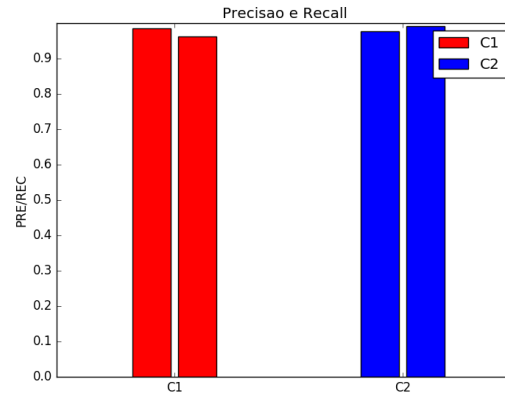


Figura 44: Precisão e Recall

## 10.4 Resultados - (10,10)

Tabela 20: Matriz de confusão

	$\hat{C}_1$ (Predita)	$\hat{C}_2$ (Predita)
$C_1$	20.30	0.90
$C_2$	0.30	35.40

- $ACC = 97.89\%$
- $ERR = 2.11\%$
- $AUC = 0.9746$
- $PRE(C1) = 0.9854$
- $REC(C1) = 0.9575$
- $PRE(C2) = 0.9752$
- $REC(C2) = 0.9916$

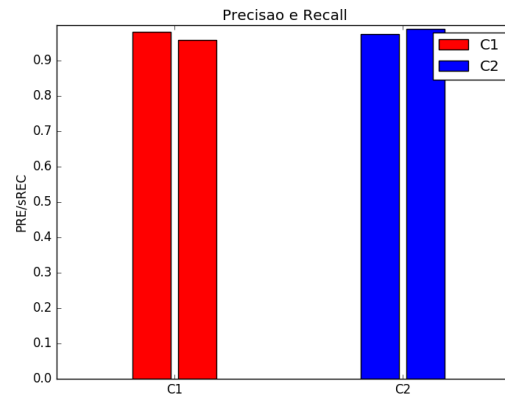


Figura 45: Precisão e Recall

## 11 Máquinas de Vetor de Suporte (SVM)

### 11.1 Formulação Matemática

O SVM baseia-se na representação indireta do espaço de características a partir do produto interno do espaço vetorial de funções.

A função de núcleo utilizada foi a RBF:  $\exp(-\gamma|x - x'|^2)$ .

O problema de otimização para o ajuste de parâmetros pode ser escrito como:

$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^n \zeta_i$$

sujeito a  $y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i$ ,  
 $\zeta_i \geq 0, i = 1, \dots, n$

$$C = \frac{N}{\alpha}$$

$w$  : direção ortogonal ao hiperplano da função discriminante

O método busca maximizar a margem de separação.

## 11.2 Resultados

A classificação por SVM é mais recomendada para vetores de características de dimensões maiores e poucos registros. No entanto, como o dataset em questão não possui uma quantidade muito grande de registros é válido utilizá-lo.

Tabela 21: Matriz de confusão

	$\hat{C}_1$ (Predita)	$\hat{C}_2$ (Predita)
$C_1$	20.40	0.80
$C_2$	0.50	35.20

- $ACC = 97.72\%$
- $ERR = 2.28\%$
- $AUC = 0.9741$
- $PRE(C1) = 0.9761$
- $REC(C1) = 0.9623$
- $PRE(C2) = 0.9778$
- $REC(C2) = 0.9860$

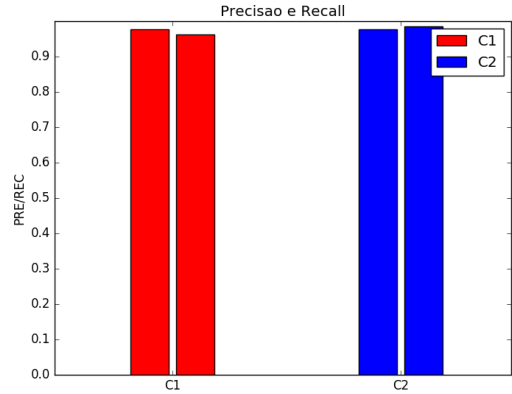


Figura 46: Precisão e Recall

## 12 Conclusão

A comparação entre os métodos pode ser vista na tabela 22, onde (NP) ou (P) indica se foi feita padronização ou não, ressaltando que os métodos MLP e SVM apresentaram problemas de convergência sem padronização.

Conforme esperado, observou-se que com o aumento de complexidade dos modelos, os resultados melhoraram. Iniciando com o Bayesiano Simples, que supoe que as variáveis não

estão correlacionadas e respeitam uma distribuição gaussiana, observaram-se resultados inferiores a todos os outros métodos. O Bayesiano Quadrático teve melhor desempenho, pois possui uma superfície de decisão quadrática, sendo portanto mais flexível. A Regressão Logística foi levemente superior, apesar de possuir uma função discriminante linear, provavelmente por utilizar o princípio da máxima verossimilhança para ajustar os parâmetros, o que é mais recomendado para problemas de classificação.

Os modelos baseados em redes neurais apresentaram os melhores resultados. Os resultados melhoravam com maior número de neurônios na rede. No entanto a adição de mais camadas não trouxe melhoras tão significativas pelo fato de o modelo não apresentar complexidade tão grande. A desvantagem é que conforme aumenta-se o número de neurônios e camadas o custo computacional cresce bastante. Além disso, o modelo é bastante sensível a padronização das variáveis. Como mostra a tabela o melhor desempenho obtido foi com o MLP com uma camada intermediária de 100 neurônios.

Tabela 22: Resultados

	<b>ACC</b>	<b>AUC</b>
Baysiano Simples (NP)	93.67%	0.9266
Baysiano Simples (P)	92.62%	0.9191
Baysiano Quadrático (NP/P)	95.78%	0.9549
Regressão Logística (NP)	95.96%	0.9563
Regressão Logística (P)	94.90%	0.9469
Perceptron (NP)	90.33%	0.8808
Perceptron (P)	96.13%	0.9615
MPL (21) (P)	97.35%	0.9708
MPL(100) (P)	98.07%	0.9769
MPL(10,10) (P)	97.89%	0.9746
SVM (P)	97.72%	0.9741

## 13 Apêndice

Tabela 23: Estatísticas básicas (todas as variáveis)

	Máximo	Mínimo	Média	Desvio Padrão	P 25	P 50	P 75
$x_1$	28.11	6.981	14.13	3.524	11.7	13.37	15.78
$x_2$	2.873	0.1115	0.4052	0.2773	0.2324	0.3242	0.4789
$x_3$	36.04	7.93	16.27	4.833	13.01	14.97	18.79
$x_4$	39.28	9.71	19.29	4.301	16.17	18.84	21.8
$x_5$	4.885	0.3602	1.217	0.5516	0.8339	1.108	1.474
$x_6$	49.54	12.02	25.68	6.146	21.08	25.41	29.72
$x_7$	188.5	43.79	91.97	24.3	75.17	86.24	104.1
$x_8$	21.98	0.757	2.866	2.022	1.606	2.287	3.357
$x_9$	251.2	50.41	107.3	33.6	84.11	97.66	125.4
$x_{10}$	2501	143.5	654.9	351.9	420.3	551.1	782.7
$x_{11}$	542.2	6.802	40.34	45.49	17.85	24.53	45.19
$x_{12}$	4254	185.2	880.6	569.4	515.3	686.5	1084
$x_{13}$	0.1634	0.05263	0.09636	0.01406	0.08637	0.09587	0.1053
$x_{14}$	0.03113	0.001713	0.007041	0.003003	0.005169	0.00638	0.008146
$x_{15}$	0.2226	0.07117	0.1324	0.02283	0.1166	0.1313	0.146
$x_{16}$	0.3454	0.01938	0.1043	0.05281	0.06492	0.09263	0.1304
$x_{17}$	0.1354	0.002252	0.02548	0.01791	0.01308	0.02045	0.03245
$x_{18}$	1.058	0.02729	0.2543	0.1573	0.1472	0.2119	0.3391
$x_{19}$	0.4268	0	0.0888	0.07972	0.02956	0.06154	0.1307
$x_{20}$	0.396	0	0.03189	0.03019	0.01509	0.02589	0.04205
$x_{21}$	1.252	0	0.2722	0.2086	0.1145	0.2267	0.3829
$x_{22}$	0.2012	0	0.04892	0.0388	0.02031	0.0335	0.074
$x_{23}$	0.05279	0	0.0118	0.00617	0.007638	0.01093	0.01471
$x_{24}$	0.291	0	0.1146	0.06573	0.06493	0.09993	0.1614
$x_{25}$	0.304	0.106	0.1812	0.02741	0.1619	0.1792	0.1957
$x_{26}$	0.07895	0.007882	0.02054	0.008266	0.01516	0.01873	0.02348
$x_{27}$	0.6638	0.1565	0.2901	0.06187	0.2504	0.2822	0.3179
$x_{28}$	0.09744	0.04996	0.0628	0.00706	0.0577	0.06154	0.06612
$x_{29}$	0.02984	0.000895	0.003795	0.002646	0.002248	0.003187	0.004558
$x_{30}$	0.2075	0.05504	0.08395	0.01806	0.07146	0.08004	0.09208