**UNIVERSIDADE ESTADUAL DE PONTA**

**DEPARTAMENTO DE INFORMÁTICA**

**ENGENHARIA DE COMPUTAÇÃO – ESTUDO DIRIGIDO**

**Mateus F. Ribeiro, Matheus H. Miranda, Paulo M. Haddad Filho**

**1) Considere um perceptron simples com 5 entradas (4 elementos de dados e um bias = 1). Suponha que o conjunto de pesos atuais é (-0.5, 0.7, 0.6, 0.1, -0.2) e que os seguintes exemplos devem ser processados:**



**Qual os valores dos pesos após o processamento destas entradas? (Assuma que a taxa de aprendizagem é de 0.5).**

(-0,5) \* 1 + 0,7 \* 1 + 0,6 \* 1 + 0,1 \* 1 + (-0,2) \* (-1) = 1,1

Saída = 1 e <W X> > 0, não altera os pesos.

(-0,5) \* 1 + 0,7 \* (-1) + 0,6 \* (-1) + 0,1 \* 1 + (-0,2) \* 1 = -1,9

Saída = 1 e <W X> <0, alteramos os pesos:

W0 = W0 + X0 \* taxa\_aprend = -0,5 + 1 \* 0,5 = 0

W1 = W1 + X1 \* taxa\_aprend = 0,7 + (-1) \* 0,5 = 0,2

W2 = W2 + X2 \* taxa\_aprend = 0,6 + (-1) \* 0,5 = 0,1

W3 = W3 + X3 \* taxa\_aprend = 0,1 + 1 \* 0,5 = 0,6

W4 = W4 + X4 \* taxa\_aprend = -0,2 + 1 \* 0,5 = 0,3

Após o processamento destas entradas, os pesos são: (0, 0.2, 0.1, 0.6, 0.3)

**2) Defina matriz de confusão e explique a formulação das medidas de acurácia, precisão e recall.**

A matriz de confusão conhecida também como matriz de erro ou tabela de confusão, é uma tabela que com duas linhas e duas colunas que relatam o número de falsos positivos, falsos negativos, verdadeiros positivos e os verdadeiros negativos. Onde permite que uma análise mais detalhada do que a mera proporção de classificações corretas (a sua precisão). A precisão para esses casos irá produzir resultados enganosos se o dado conjunto de dados estiver desiquilibrado, sendo que, quando o número de observações em diferentes classes varia muito.

Simplificando, a matriz de confusão nada mais é que uma matriz n por n, sendo n definido como o número de classes disponíveis para um modelo classificar. Para cada classificação que for realizada, é incrementado o valor em uma das células da matriz (x linhas e y colunas), x é definida como corrente do classificador, e y é definida como classificação prevista (seu valor real). Onde esperamos que as classificações corretas se encontrem na diagonal principal da matriz.

Simplificando para a precisão, ela é relacionada aos acertos positivos do conjunto, a sua fórmula definida por: verdadeiros positivos (VP) fracionado pela soma dos verdadeiros positivos (VP) com os falsos negativos (FN). (VP/(VP+FN))

O Recall é definido pela precisão, a sua diferença da precisão é de que é direcionado para apenas uma classe específica.

A acurácia, como em estatística, é a taxa de acerto geral do algoritmo, sendo então identificada pela soma de acertos positivos (AP) com acertos negativos (AN) fracionada pelo total multiplicado pela soma de verdadeiros positivos (VP) com os verdadeiros negativos (VN) fracionado pelo total. Ou seja: (AP+AN) / Total \*((VP + VN) / Total)

**3) Descreva os seguintes elementos do treinamento de uma rede neural multicamadas e o efeito esperado de sua definição sobre o desempenho do MLP:**

**a) Número de época**

O número de época é o valor da iteração atual de um treinamento de um dado modelo, onde uma iteração se completa quando ocorre de todos os exemplos da base de dados inicial que passaram pelo treinamento uma única vez.

**b) Taxa de aprendizagem**

A taxa de aprendizagem é um fator multiplicativo que ocorre na fase de atualização dos pesos do conjunto, tal que quanto maior essa taxa em que se aprende, maior é o alcance da atualização dos pesos, ou seja, a cada passo ocorrido no treinamento acaba influenciando sempre mais no resultado e fazem com que os valores atribuídos aos pesos flutuem mais.

**c) Momentum**

O momentum ou momento é um fato que pode muito impactar diretamente na taxa de aprendizagem, onde que, espera-se a cada época a taxa de aprendizado reduza seu valor para que seja possível divergir para pesos com ajustes mais precisos ou finos, sendo defino como um fator redutor para a taxa de aprendizagem a fim de melhorias.

**d) Normalização das entradas**

A normalização de entradas é um processo que ocorre antes do treinamento ser realizado, tal que os valores de entrada passam por uma função que atualiza seus valores a fim de manter um padrão de dados, onde é muito mais comum ocorrer a normalização entre 0 e 1.

**4) Considere archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(28diagnostic), utilize o software WEKA para treinar uma rede neural de múltiplas camadas para classificar os casos contidos na base de dados. Use redes com uma camada intermediária, uma camada de entrada e uma cada de saída. Durante os testes, utilize redes com as seguintes quantidades de nós na camada intermediária: 10, 15 e 20 . No treinamento de cada configuração rede realize testes com 200, 400 e 600 épocas.**

**a) Para cada um dos testes determine qual das seguintes taxas de aprendizagem teve o melhor desempenho : (0.1, 0.3, 0.5)**

Camadas Intermediárias: 10, 15 e 20

Número de Épocas: 200 e 400 e 600

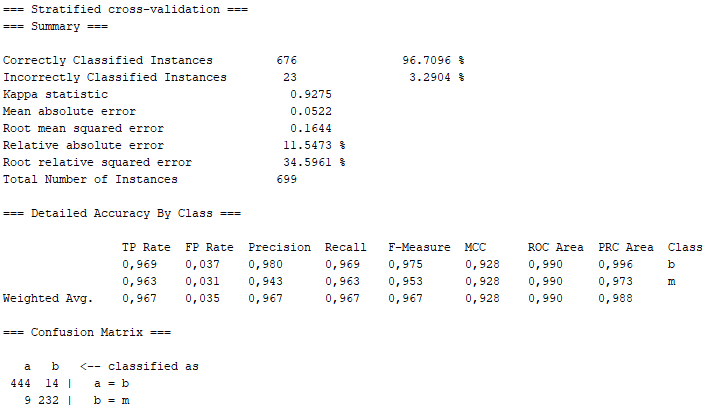
Taxas de aprendizagem: 0.1, 0.3 e 0.5

Resultados utilizando validação cruzada com 10 folds:

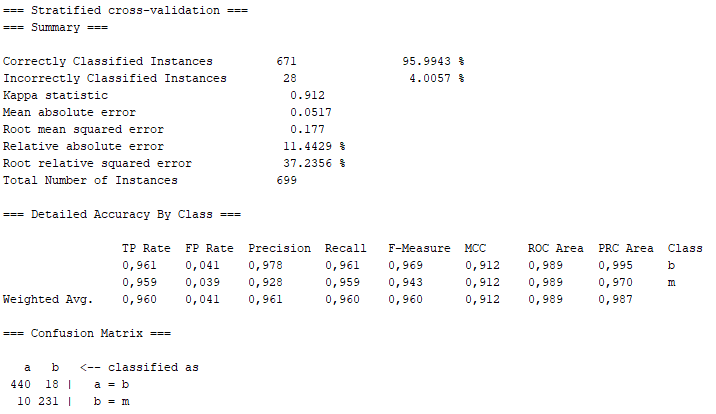
Os valores nos itens indicam, respectivamente:

- Camadas Intermediárias x Número de Épocas x Taxa de Aprendizagem:

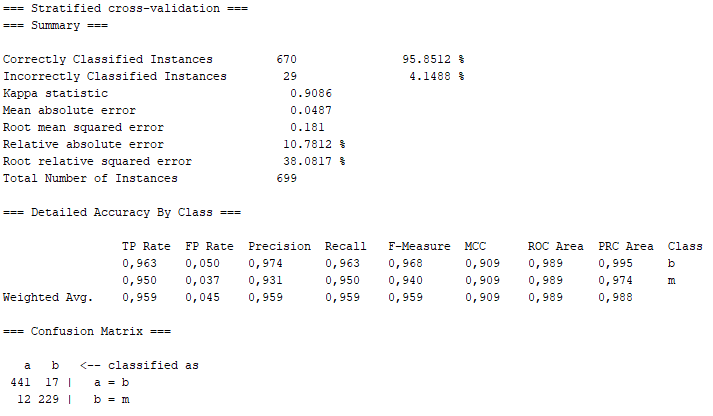
- 10 x 200 x 0.1:



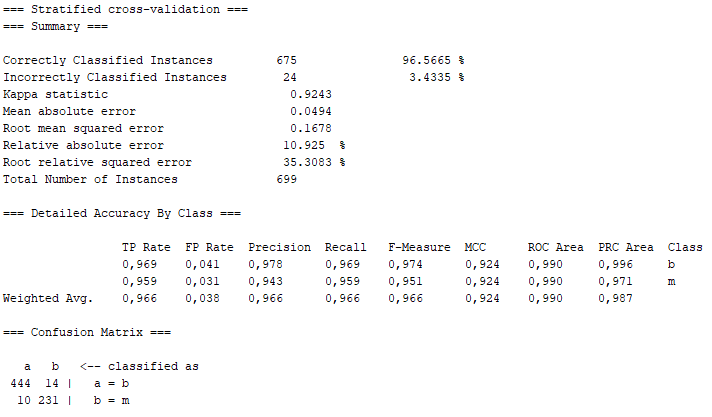
- 10 x 400 x 0.1:



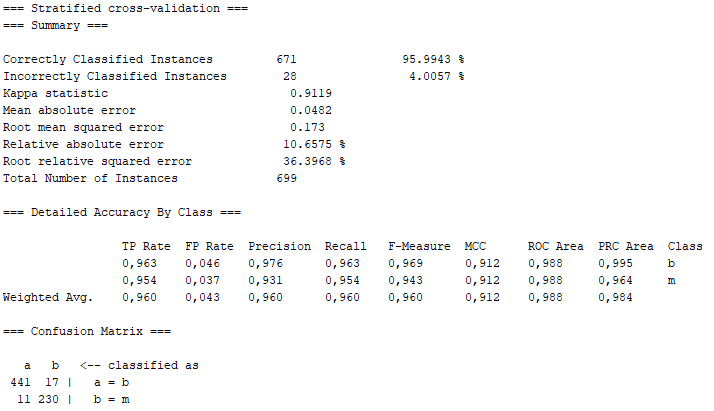
- 10 x 600 x 0.1:



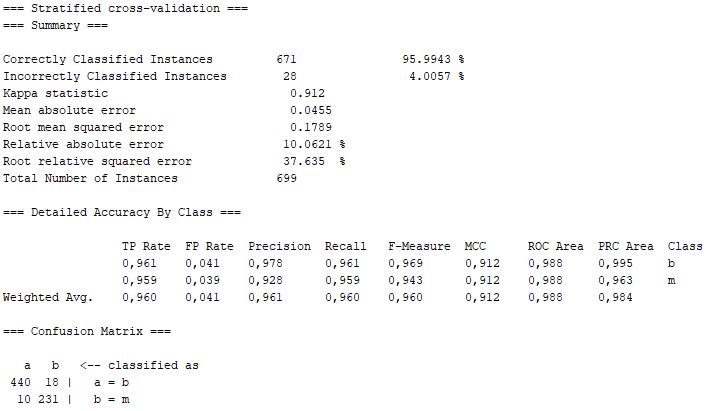
- 15 x 200 x 0.1:



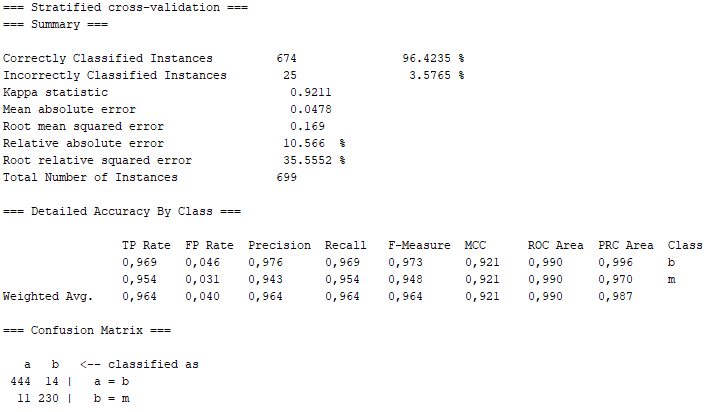
- 15 x 400 x 0.1:



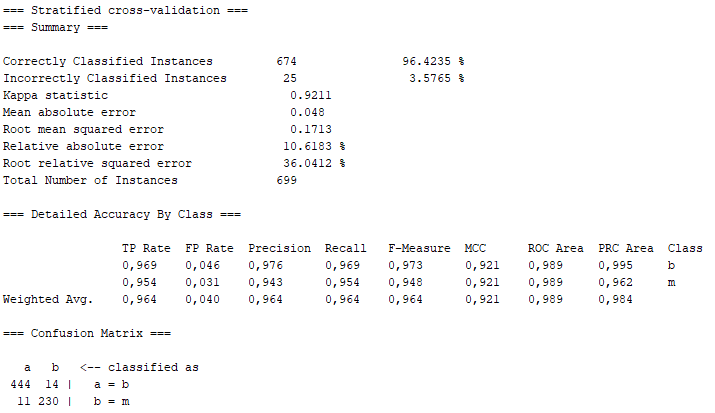
- 15 x 600 x 0.1:



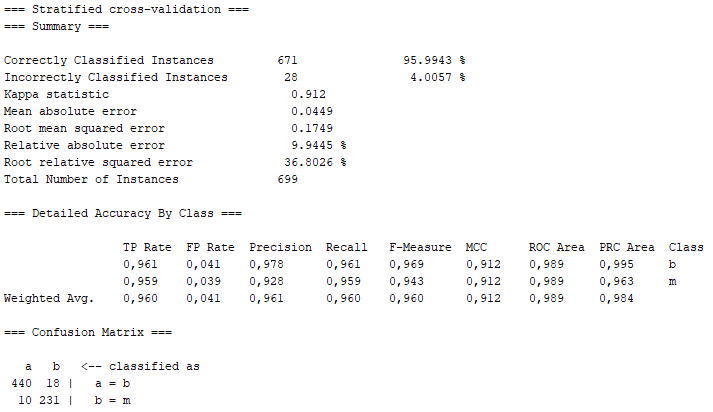
- 20 x 200 x 0.1:



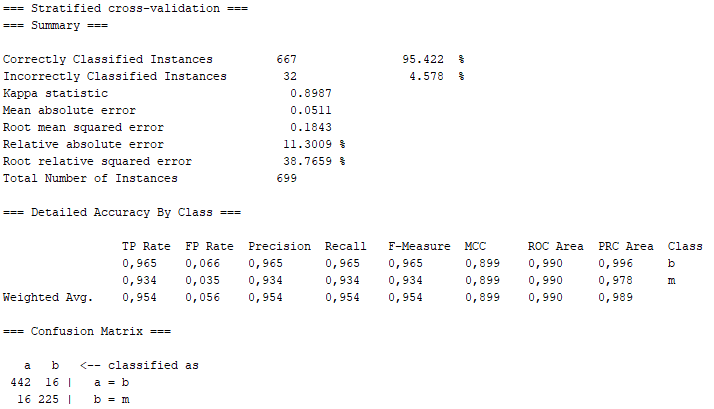
- 20 x 400 x 0.1:



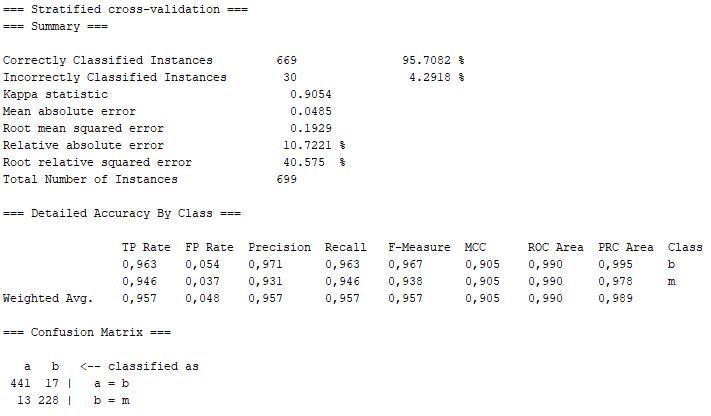
- 20 x 600 x 0.1:



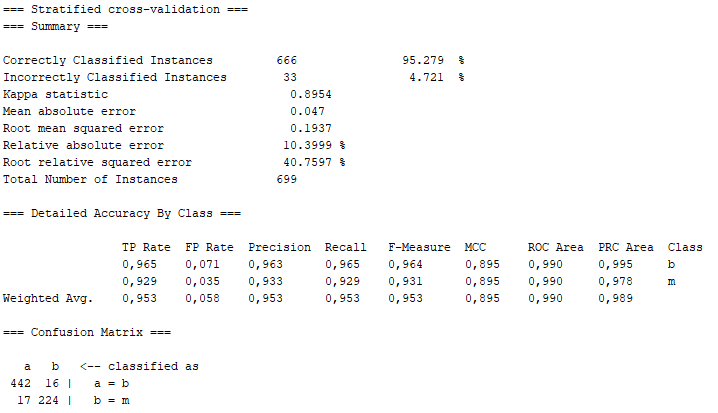
- 10 x 200 x 0.3:



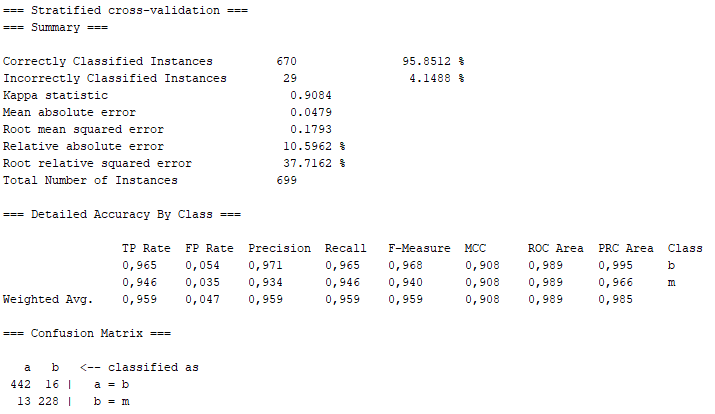
- 10 x 400 x 0.3:



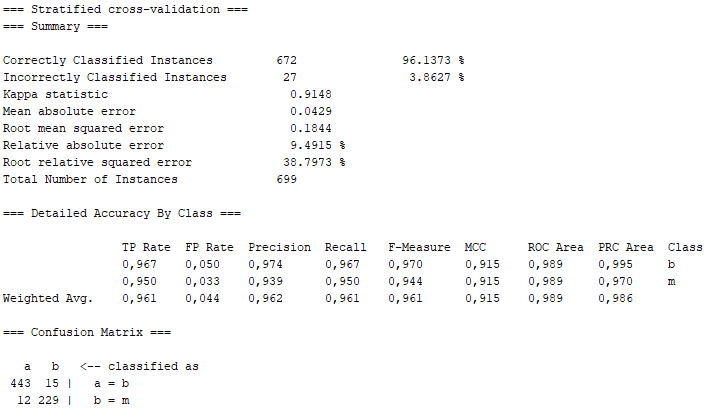
- 10 x 600 x 0.3:



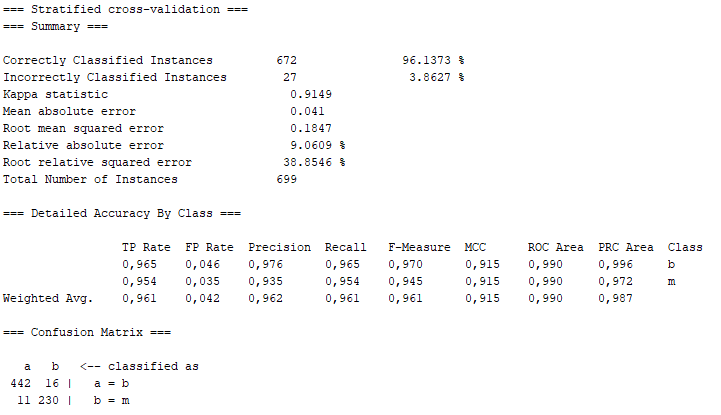
- 15 x 200 x 0.3:



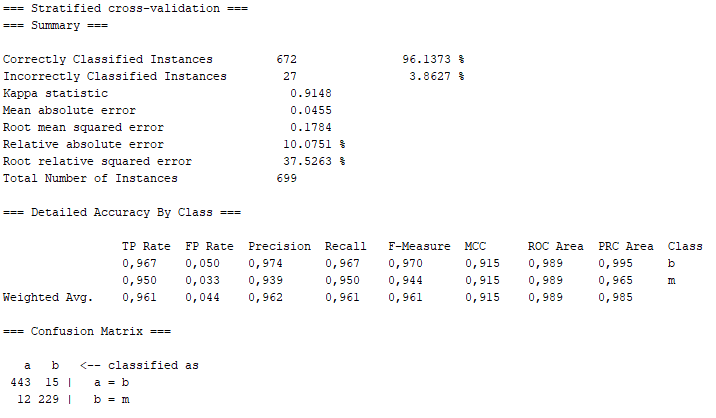
- 15 x 400 x 0.3:



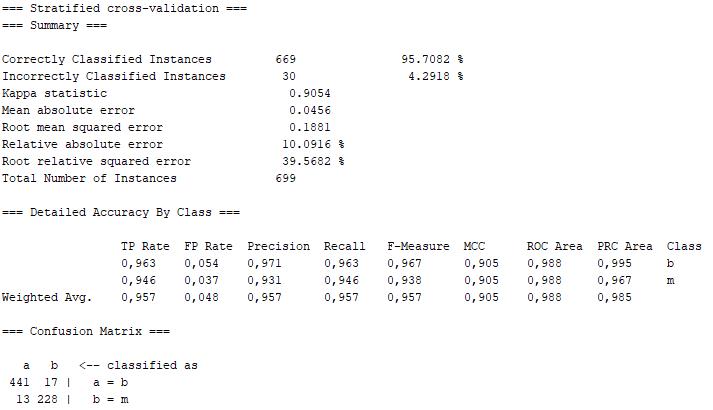
- 15 x 600 x 0.3:



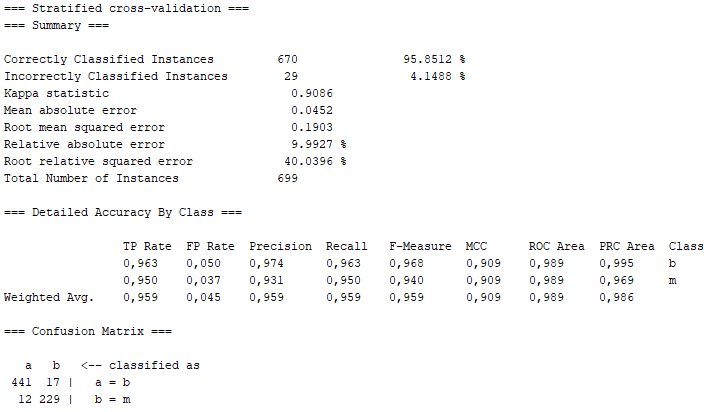
- 20 x 200 x 0.3:



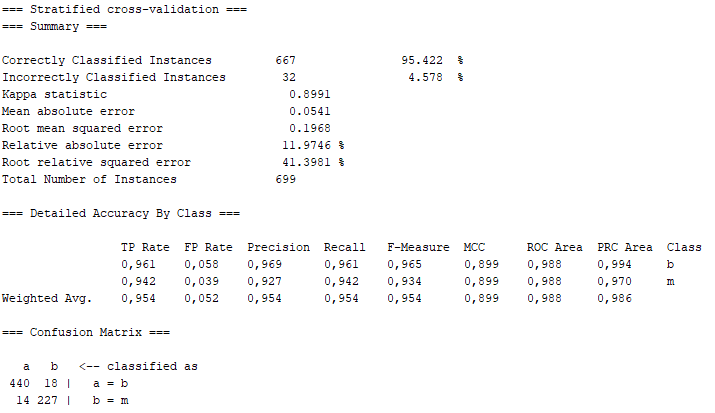
- 20 x 400 x 0.3:



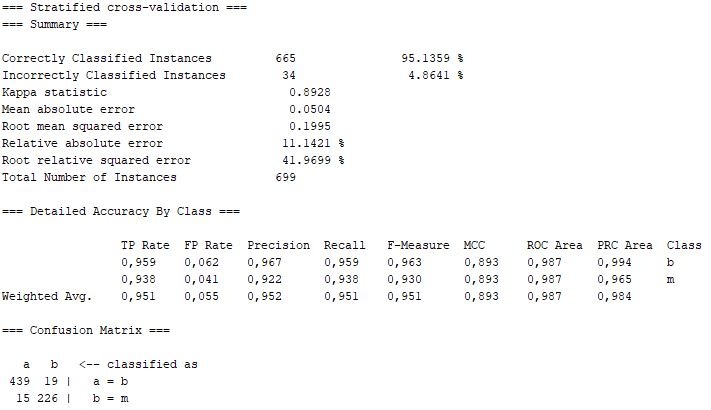
- 20 x 600 x 0.3:



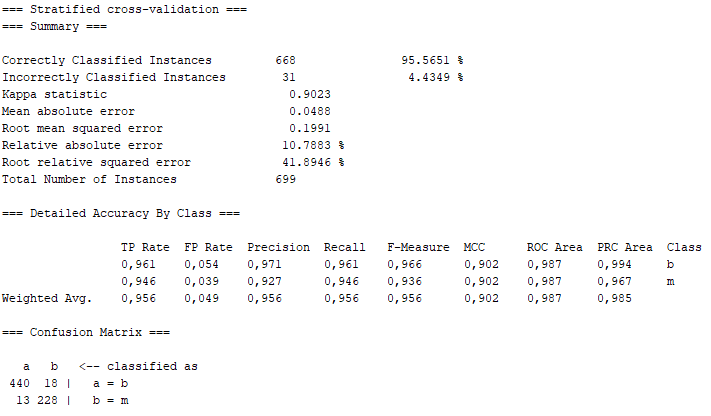
- 10 x 200 x 0.5:



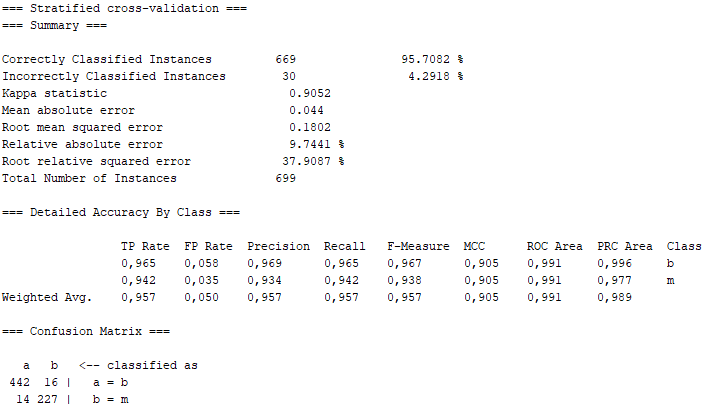
- 10 x 400 x 0.5:



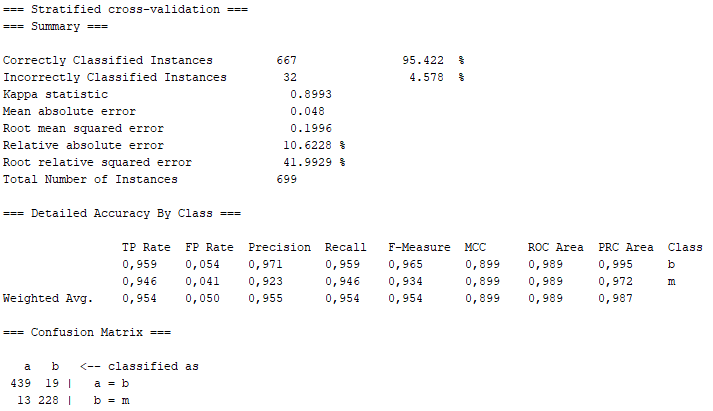
- 10 x 600 x 0.5:



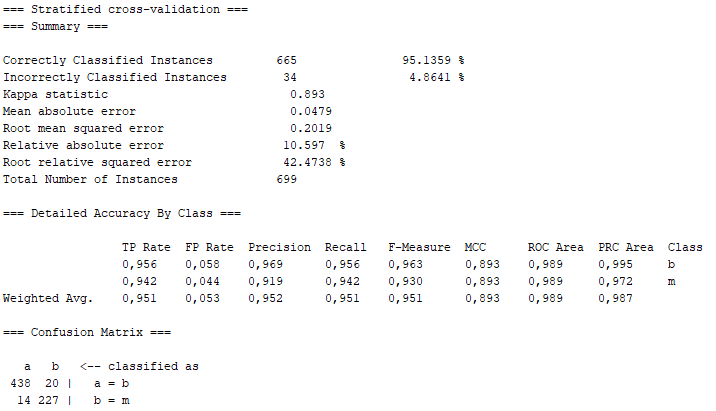
- 15 x 200 x 0.5:



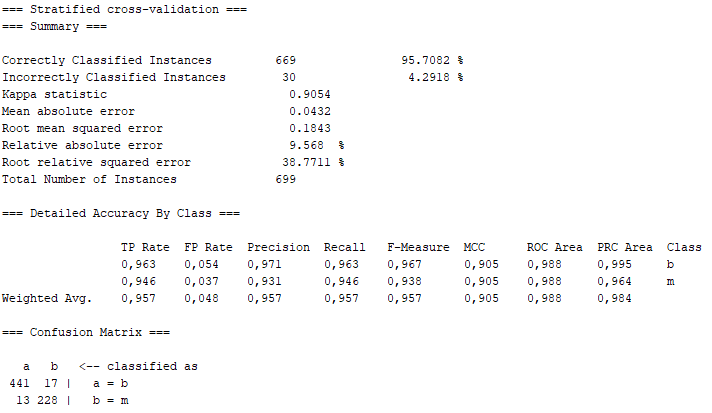
- 15 x 400 x 0.5:



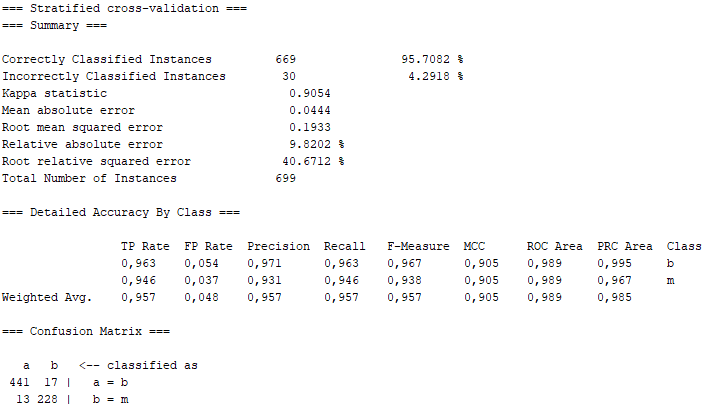
- 15 x 600 x 0.5:



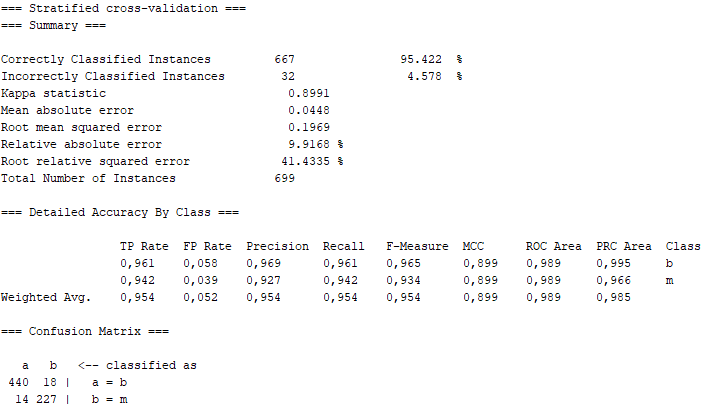
- 20 x 200 x 0.5:



- 20 x 400 x 0.5:



- 20 x 600 x 0.5:



O melhor desempenho foi com a Taxa de Aprendizagem de 0.3, com uma média de acurácia de 96,3916 %.

O melhor desempenho para cada um dos testes foi com as seguintes taxas de aprendizagem:

- 10 x 200: 0.3

- 10 x 400: 0.3

- 10 x 600: 0.3

- 15 x 200: 0.3

- 15 x 400: 0.3

- 15 x 600: 0.3

- 20 x 200: 0.3

- 20 x 400: 0.3

- 20 x 600: 0.3

**b) Qual rede teve a maior acurácia?**

5 configurações de rede apresentaram o maior valor de acurácia, sendo estas:

(Nº camadas intermediárias x Nº épocas x Taxa de aprendizagem : Acurácia)

- 15 x 200 x 0.3: 96,5665 %

- 15 x 400 x 0.3: 96,5665 %

- 15 x 600 x 0.3: 96,5665 %

- 20 x 400 x 0.3: 96,5665 %

- 20 x 600 x 0.3: 96,5665 %

**c) Qual o efeito do número de épocas sobre o treinamento, do número de nós na camada intermediária e da taxa de aprendizagem no desempenho da rede em termos de erro e acurácia?**

Um número alto de épocas pode aumentar a acurácia, pois fará o algoritmo processar os pesos por um período maior, porém se aumentarmos muito esse número, podemos chegar em um ponto onde a acurácia já estabilizou, ou seja, vai aumentar o processamento do treinamento sem melhorar a acurácia. Um número muito baixo de épocas pode não ser o suficiente para o algoritmo encontrar um peso bom, por exemplo, se a taxa de aprendizagem for baixa, o que faz os pesos serem alterados mais lentamente, diminuindo a acurácia.

Um número alto de nós na camada intermediária pode aumentar a acurácia, porém ele pode chegar em um ponto onde aumentamos o número de nós e a acurácia não aumenta, pois ela chegou em um ponto onde fica estabilizada, o que vai aumentar o processamento do treinamento desnecessariamente. Um número muito baixo de nós pode piorar o cálculo dos novos pesos, fazendo a acurácia diminuir.

A taxa de aprendizagem influencia na velocidade com que os pesos são alterados. Se ela for muito baixa, a precisão do ajuste de pesos aumenta, porém os pesos vão demorar muito para serem corrigidos para um valor ótimo, o que pode diminuir a acurácia ou exigir um número muito grande de épocas. Se ela for muito alta, os pesos sofrerão maior alteração, o que pode diminuir a precisão do ajuste de pesos e levar a uma situação onde o algoritmo nunca consegue encontrar um peso bom, o que pode diminuir muito a acurácia.

**d) Interprete a qualidade do melhor resultado obtido em termos de acurácia, precisão e recall.**

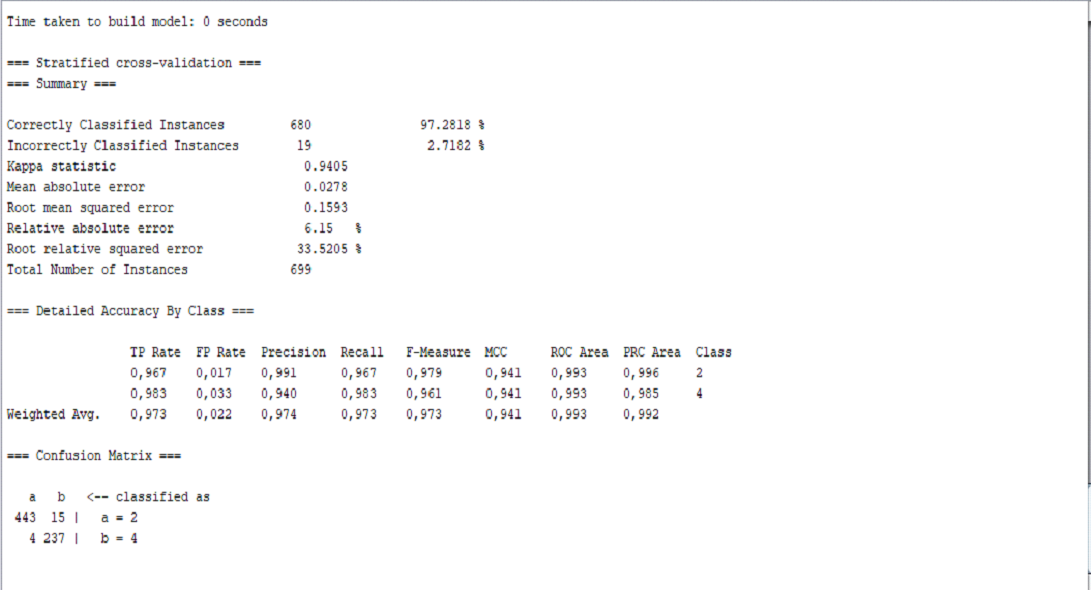
..............................................................

**5) Compare o desempenho da melhor rede obtida no exercício 2 com aquele do classificador Naive Bayes (use o WEKA). Interprete os resultados em termos de acurácia, precisão e recall. Oque significam os parâmetros de cada modelo?**

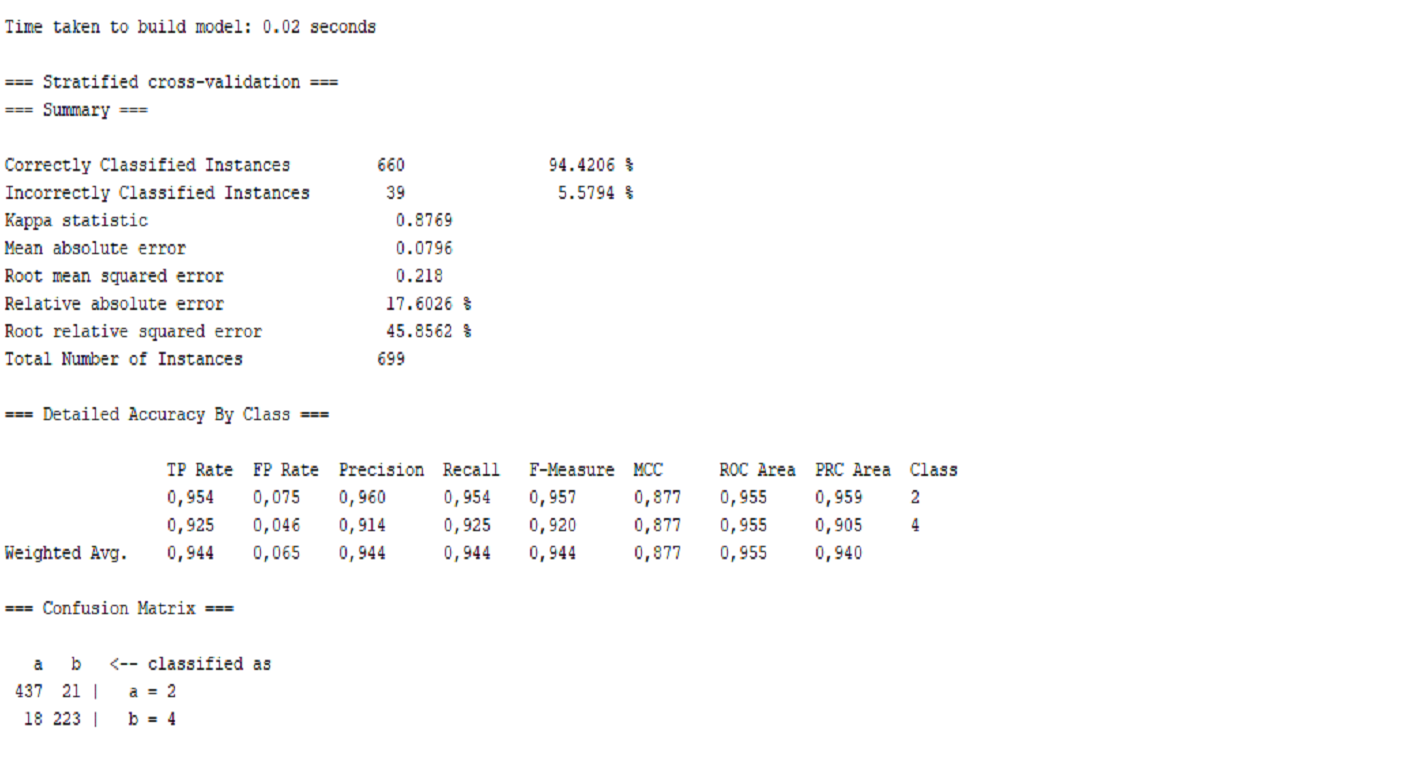
Melhor classificador do exercício 4:

Acurácia: 96,5665 %

Classificador Naive Bayes:



Classificador J48:



O melhor classificador foi o Naive Bayes, com acurácia de 97,2818 %, seguido da rede neural de múltiplas camadas, com 96,5665 %, e por último ficou o J48, com 94,4206 %.

**6) Leia a descrição do arquivo de dados semeion.csv no diretório UCI** [**http://archive.ics.uci.edu/ml/datasets/Artificial+Characters**](http://archive.ics.uci.edu/ml/datasets/Artificial+Characters)**. O arquivo sprimos.csv (anexo a esta atividade) foi gerado a partir daquele substituindo-se as saídas S1..S10 por uma única saída S1 que informa se o número é um primo ou não. Utilize o software WEKA para treinar redes neurais para a classificação dos objetos deste arquivo.**

**Durante os testes considere combinações de:**

**a) 3 diferentes taxas de aprendizado**

**b) 2 configurações da camada intermediária (máximo 15 nós)**

**d) 3 configurações do número de épocas (>200 e <1000)**

**Interprete os resultados em termos de acurácia, precisão e recall.**

Taxas de aprendizagem escolhidas: 0,5 e 1 e 0,75

Camadas Intermediárias escolhidas: 6 e 12

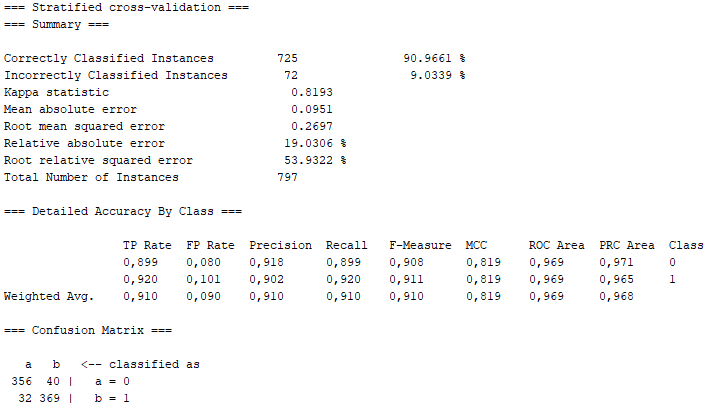
Número de Épocas escolhidas: 200 e 500 e 800

Resultados utilizando validação cruzada com 10 folds:

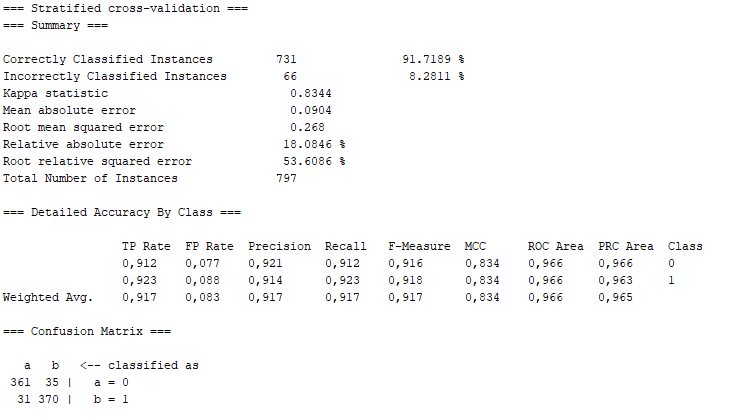
Os valores nos itens indicam, respectivamente:

- Taxa de Aprendizagem x Camadas Intermediárias x Número de Épocas:

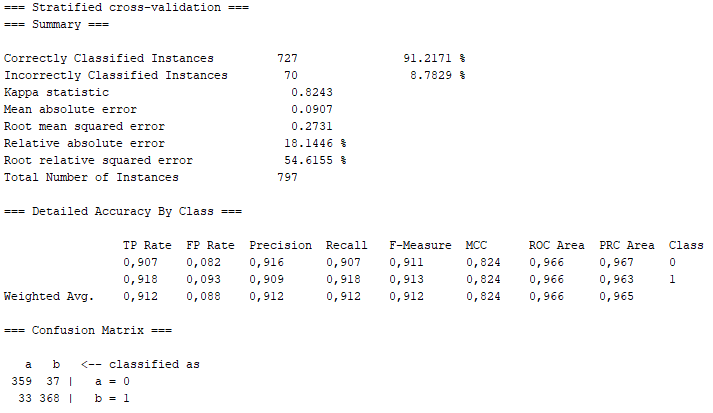
- 0,5 x 6 x 200:



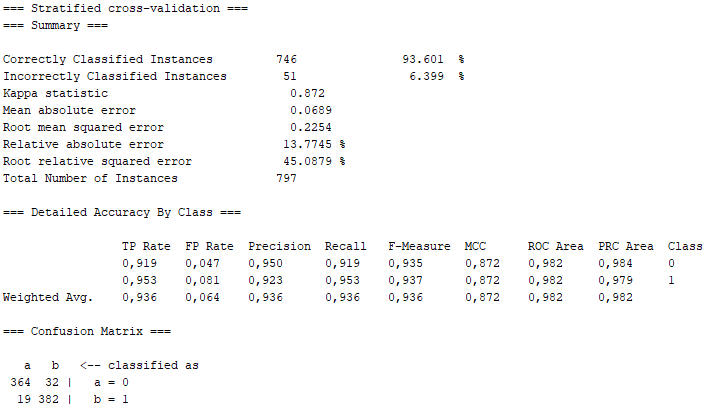
- 0,5 x 6 x 500:



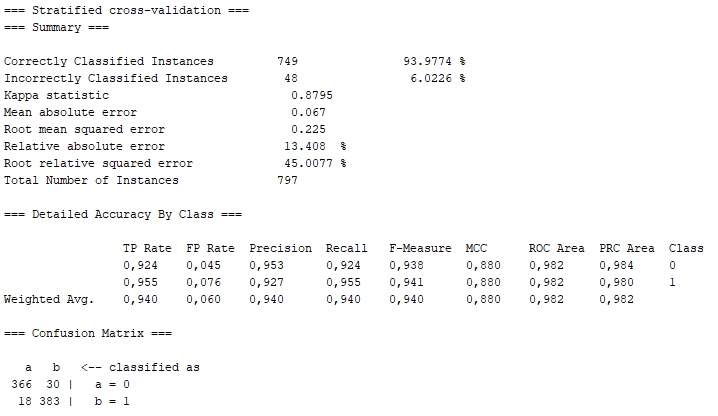
- 0,5 x 6 x 800:



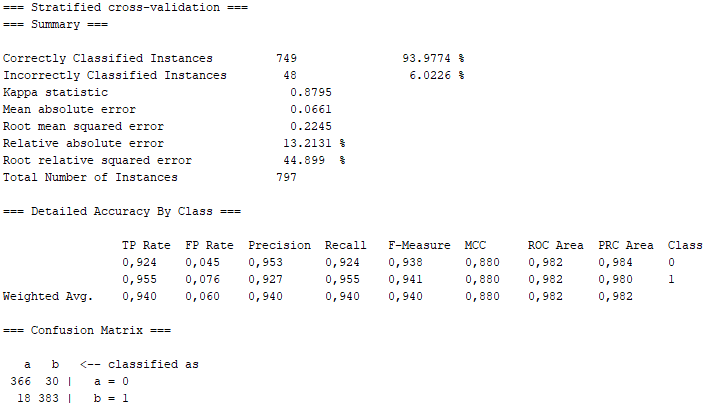
- 0,5 x 12 x 200:



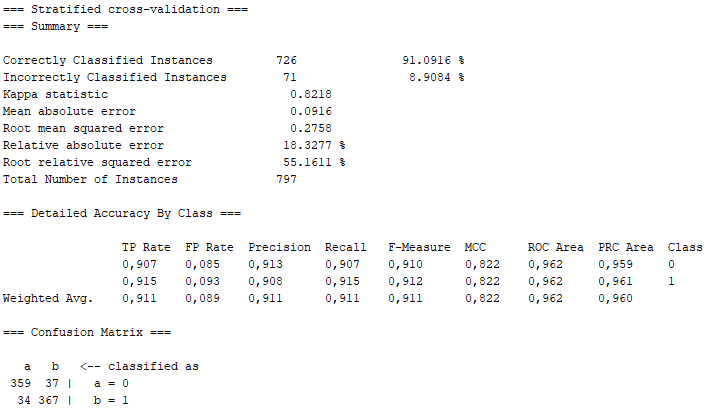
- 0,5 x 12 x 500:



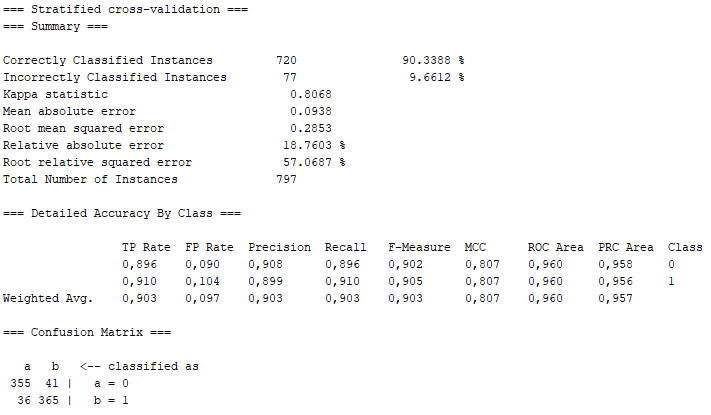
- 0,5 x 12 x 800:



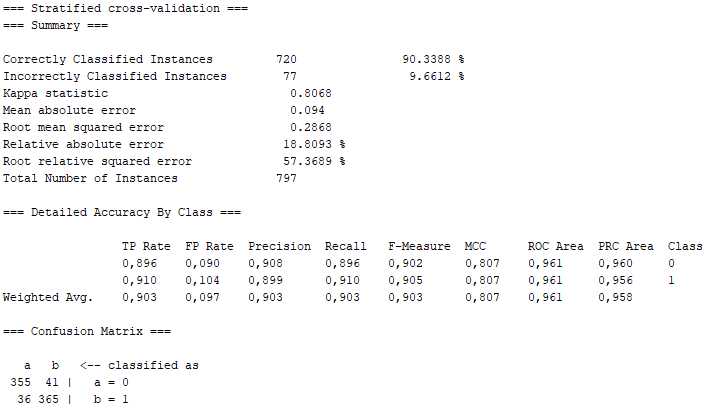
- 1 x 6 x 200:



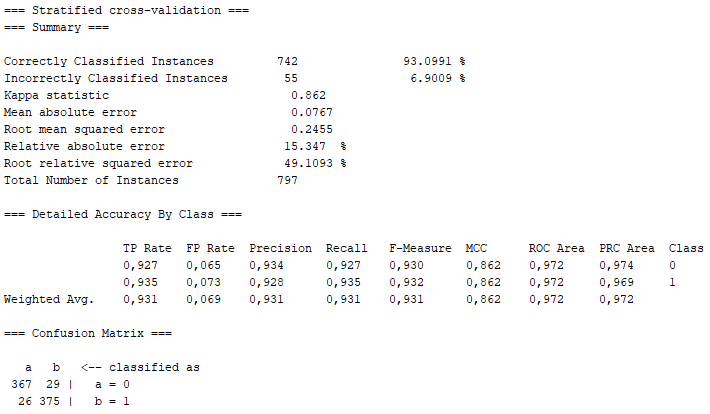
- 1 x 6 x 500:



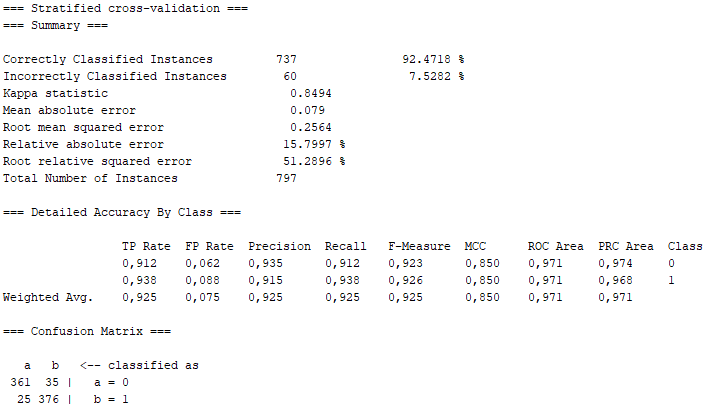
- 1 x 6 x 800:



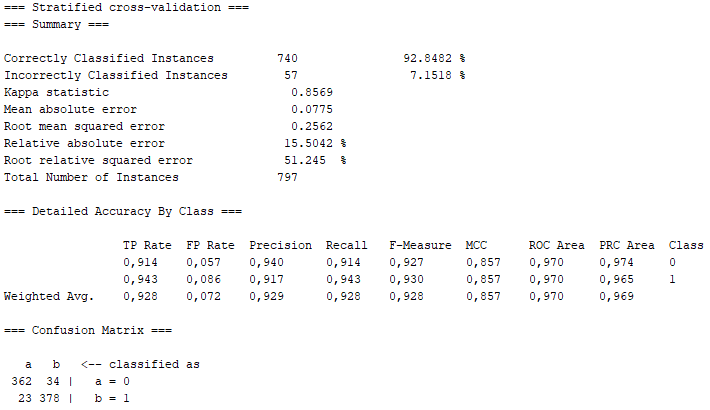
- 1 x 12 x 200:



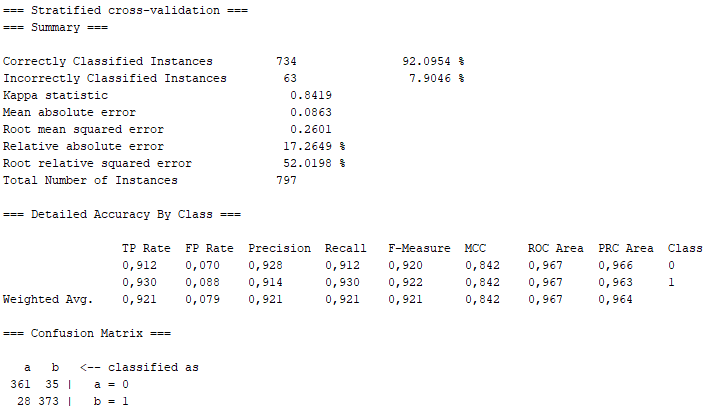
- 1 x 12 x 500:



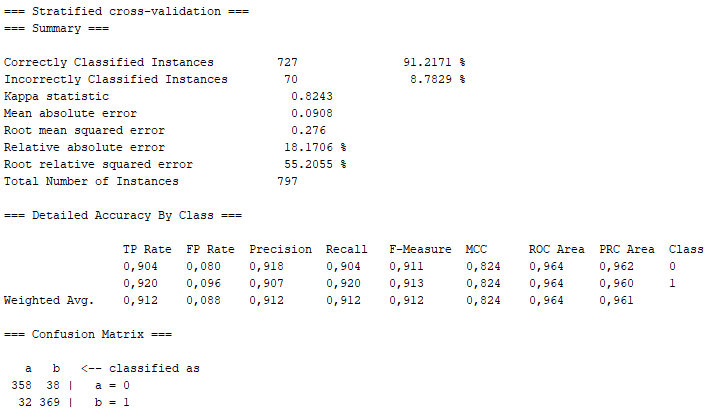
- 1 x 12 x 800:



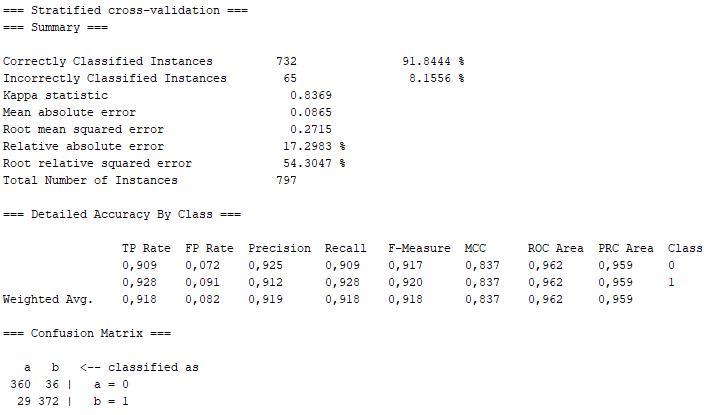
- 0,75 x 6 x 200:



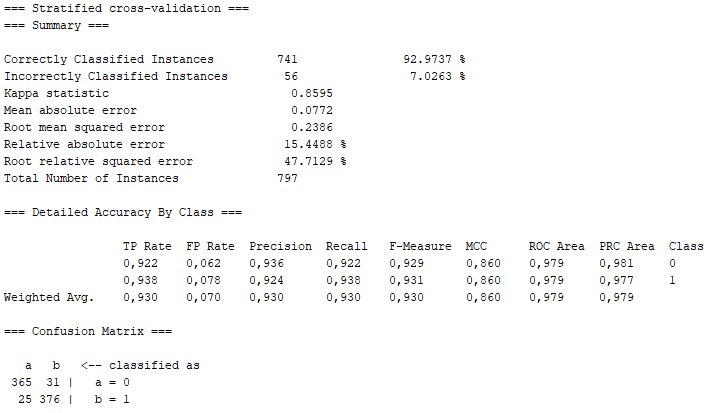
- 0,75 x 6 x 500:



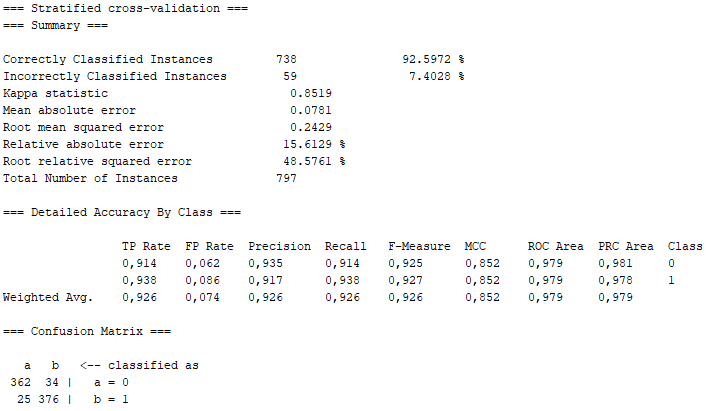
- 0,75 x 6 x 800:



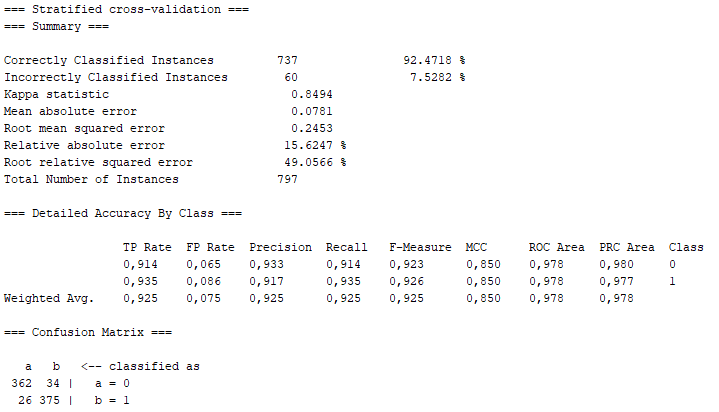
- 0,75 x 12 x 200:



- 0,75 x 12 x 500:



- 0,75 x 12 x 800:



Foi obtido dois classificadores com o melhor desempenho treinado, sendo estes:

- Taxa de Aprendizagem: 0.5

- Camadas Intermediárias: 12

- Número de Épocas: 500 e 800

A acurácia destes classificadores foi de 93,9774 %.

**7) O que é o teste de Friedman? Ele pode ser usado para comparar o desempenho de algoritmos de aprendizagem? Como?**

O teste de Friedman é um teste estatístico não-paramétrico. Semelhante ao Anova, é utilizado para detectar diferenças nos tratamentos em vários experimentos de teste. O procedimento envolve a classificação de cada linha/bloco, então considerando os valores dos postos de colunas. O teste também é usado para medidas repetidas de análise unidirecional de variância dos postos. Sendo seu uso semelhante ao Kruskal-Wallis por postos.

Sim o teste de Friedman consegue comparar desempenho de algoritmos de aprendizagem. Em outras palavras, o teste de Friedman é utilizado para comparar dados amostrais vinculados, ou seja, quando o mesmo individuo é avaliado mais de uma vez. O método não utiliza os dados numéricos diretamente, mas sim os postos ocupados por eles após a ordenação feita para cada grupo separadamente. Após a ordenação é testada a hipótese de igualdade da soma dos postos de cada grupo.

É útil quando é necessário utilizar três ou mais situações experimentais. Deve ser utilizado para um design relacionado quando os mesmos sujeitos são distribuídos por três ou mais situações experimentais. Sendo que uma vez se trata de um design relacionado no qual o mesmo sujeito obtém resultados em todas as situações, é permitido comparar os resultados de cada sujeito através de todas as situações, no sentido de verificar em que situação obtém maiores e menores resultados.

Em questão de algoritmos, quando temos os dados de k amostras correspondentes se apresentam com pelo menos uma escala de ordem crescente ou decrescente, para comprovar a hipótese de nulidade, onde que as k amostras tenham sido extraídas da mesma população. Até que a correspondência possa ser estabelecida, quando se realiza o estudo de um mesmo grupo de indivíduos sob cada uma das k condições. Para identificar alguma diferença entre as medianas é estatisticamente significativa, realiza-se a comparação do valor-p com o seu nível de significância com o intuito de avaliar a hipótese nula. A hipótese nula afirma que as medianas populacionais são todas iguais.