**Marcelo Paulon - 1411029**

INF1399 - Descoberta do Conhecimento – 2ª. Lista

1. Carregue o arquivo supermercado.csv no Weka:
2. Quantas regras existem com suporte igual a 10% e confiança igual a 90% (default)? **Zero (nenhuma regra)**.
3. Altere o suporte do item “a” para 3%. Quais foram as melhores regras de associação encontradas?

**1. Laranja=S, Maca=S, Alho=S ==> Limao=S**

**2. Limao=S, Tomate=S, Alho=S ==> Kiwi=S**

**3. Kiwi=S, Cebola=S, Pepino=S, Alho=S ==> Maca=S**

1. Altere a confiança do item “a” para 50%. Quais foram as melhores regras de associação encontradas?

**1. Laranja=S ==> Banana=S**

**2. Laranja=S, Limao=S ==> Banana=S**

**3. Maca=S, Limao=S ==> Banana=S**

**4. Banana=S, Maca=S ==> Limao=S**

**5. Limao=S, Alface=S ==> Banana=S**

**6. Limao=S ==> Banana=S**

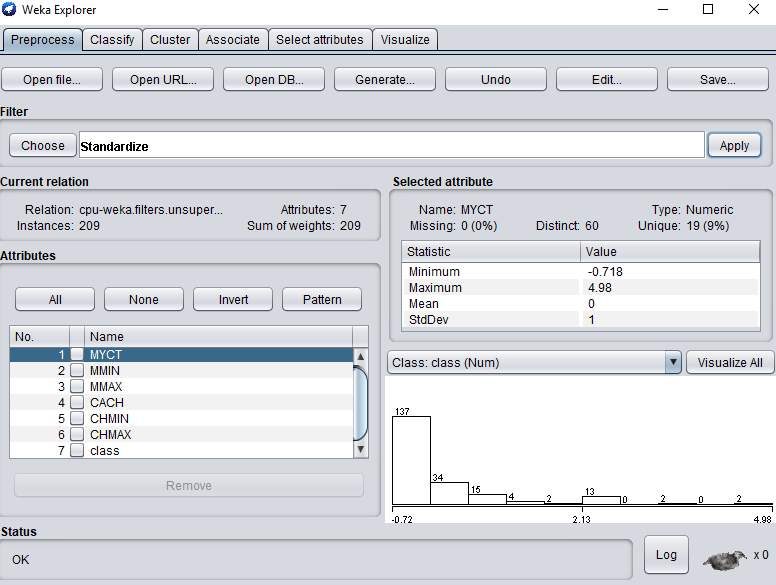
**7. Banana=S, Limao=S ==> Alface=S**

**8. Tomate=S ==> Alface=S**

**9. Alface=S ==> Banana=S**

**10. Alho=S ==> Kiwi=S**

1. Carregue o arquivo cpu.arff no Weka:
2. Normalize os atributos usando Z-score (observe que não é usando Min-max);



1. Explore o algoritmo k-means (SimpleKMeans) e calcule os clusters pré-definidos pelo algoritmo.

**Clustered Instances**

**#0 171 (82%)**

**#1 38 (18%)**

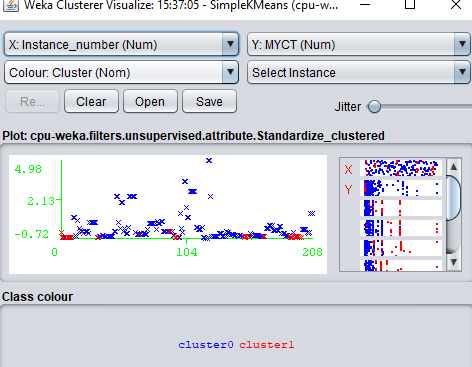
**(Captura de tela com o modelo disponível na próxima página)**

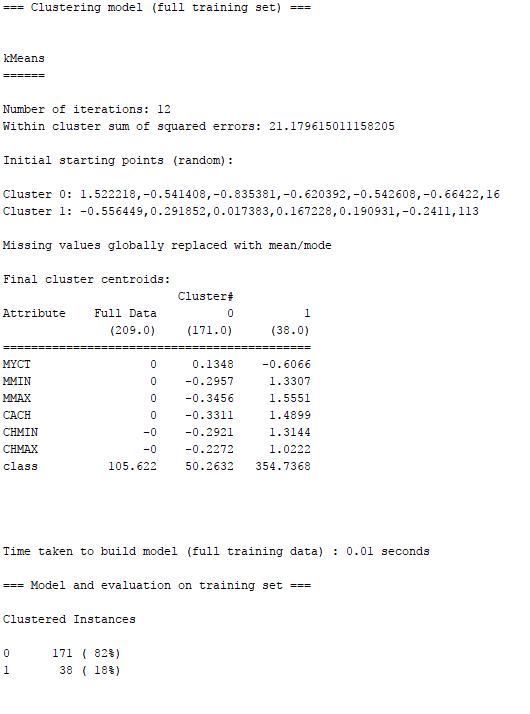
1. Mude a semente (seed) para o k-means e observe o comportamento do algoritmo. O que ocorreu? Apresente telas de associação dos clusters para demonstrar o que ocorreu.

**Clustered Instances**

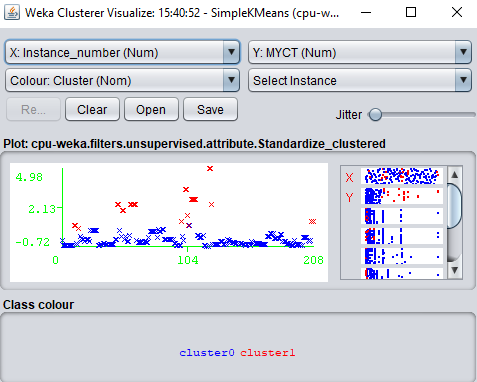
**#0 182 ( 87%)**

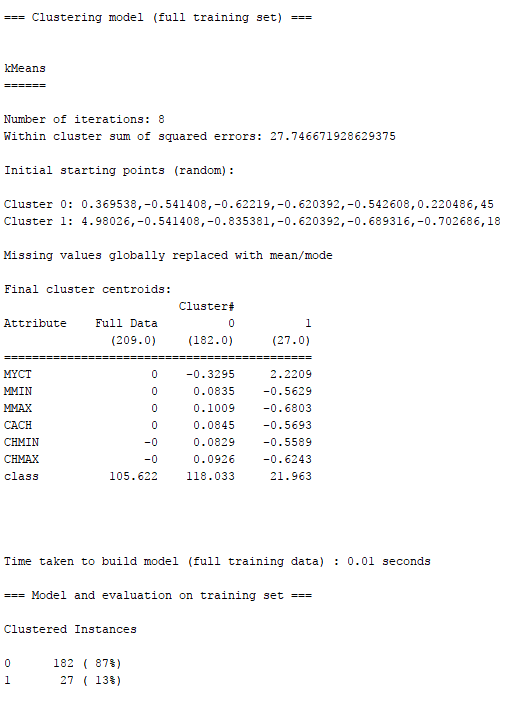
**#1 27 ( 13%)**





**Seed original (10)**



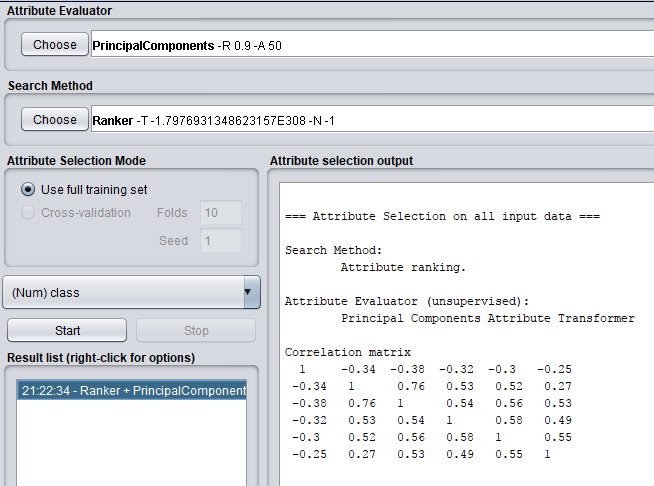


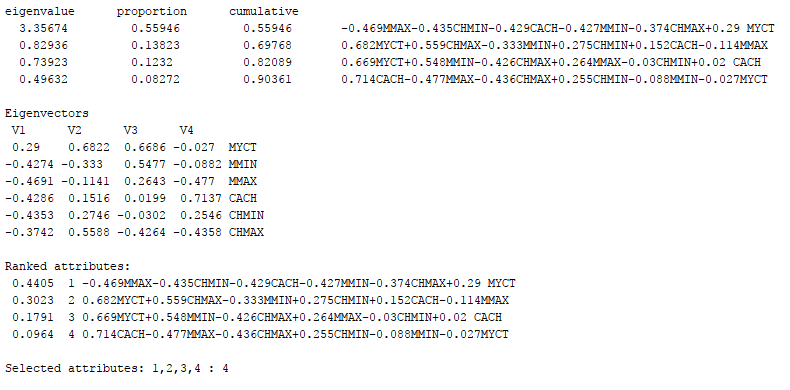
**Seed alterada (56)**

**Na segunda execução, o resultado foi diferente pois os clusters foram inicializados de forma diferente:**

**“Seed is just a random numbers seed. Once seed is fixed, even a randomized algorithm behaves deterministically. KMeans is not deterministic, so if you want repeatable results - you fix a seed. However there is completely no relation between exact value of the seed and the results of KMeans clustering. … the *only* purpose of seeding is to make sure that you get the exact same result when you run this code many times on the exact same data. KMeans is randomized, and so could lead to many different results if you simply run it many times - this way you can "force it" to be repeatable” (**<https://stackoverflow.com/questions/33973817/what-is-the-seed-in-wekas-simplekmeans-clusterer>**)**

1. Quantos dos atributos gerados através da Análise de Componentes Principais (PCA) devem ser utilizados considerando uma cobertura de 90% da variância dos dados? Qual a composição destes em relação aos atributos originais?





**Output da execução do PCA considerando uma cobertura de 90% da variância dos dados**

**Como percebemos pela imagem acima, devem ser utilizados 4 atributos. A composição dos 4 atributos em relação aos atributos originais é, como mostra a imagem:**

**Atributo 1:**

**-0.469\*MMAX-0.435\*CHMIN-0.429\*CACH-0.427\*MMIN-0.374\*CHMAX+0.29\*MYCT**

**Atributo 2:**

**0.682\*MYCT+0.559\*CHMAX-0.333\*MMIN+0.275C\*HMIN+0.152\*CACH-0.114\*MMAX**

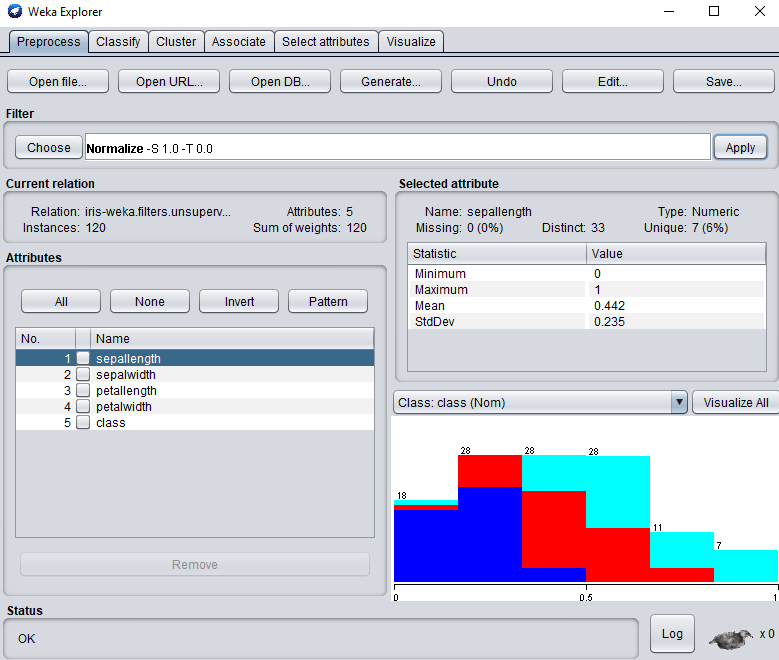
**Atributo 3**

**0.669\*MYCT+0.548\*MMIN-0.426\*CHMAX+0.264\*MMAX-0.03\*CHMIN+0.02\*CACH**

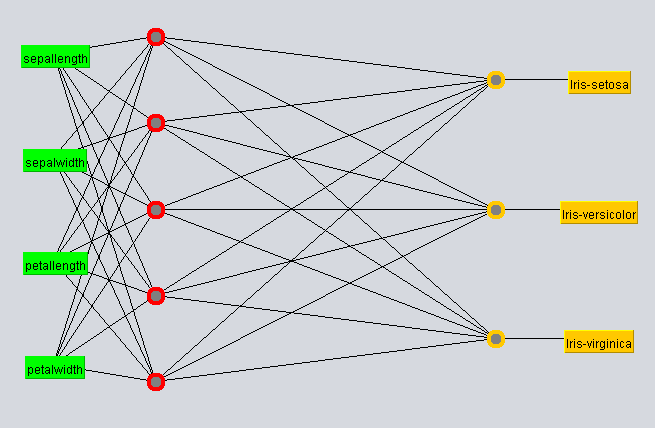
**Atributo 4**

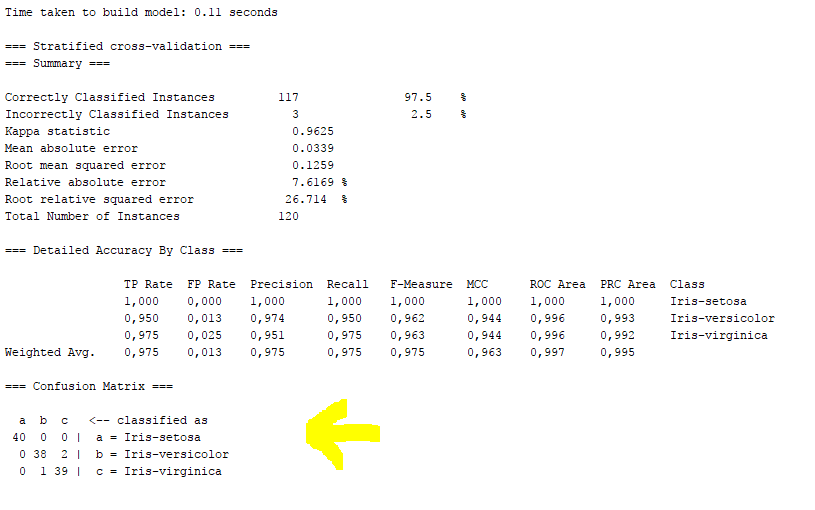
**0.714\*CACH-0.477\*MMAX-0.436\*CHMAX+0.255\*CHMIN-0.088\*MMIN-0.027\*MYCT**

1. Carregue o arquivo iris-train.arff no Weka:
2. Normalize os atributos usando Min-max;

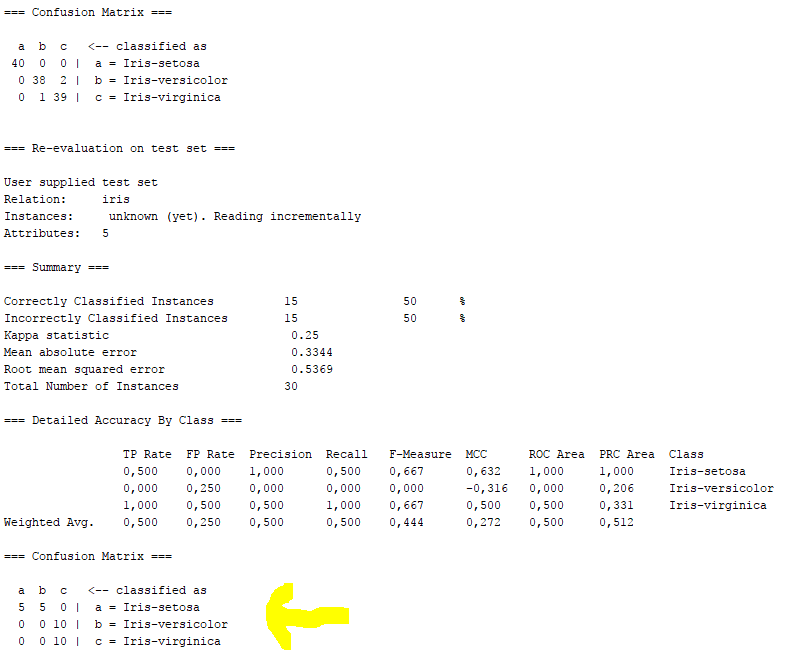


1. Treine uma RNA para classificação utilizando os dados no item a). A rede deverá ter 5 neurônios na camada escondida. Entregue como resultado a imagem da rede obtida no Weka e a matriz de confusão resultante. Treine utilizando a opção Cross-validation com valor padrão.



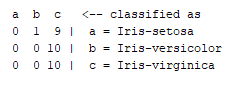
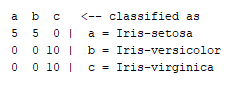


1. Submeta o arquivo iris-test.arff à rede treinada no item b) e avalie o resultado obtido na classificação apresentando a matriz de confusão.



1. Reduza o número de épocas para 10 e faça um novo treinamento da rede. O que ocorreu?

**O modelo resultante foi pior que o modelo original com 500 épocas. A rede treinada com cross validation passou a acertar apenas 83.33% (contra 97.5% de antes) e a aplicação dos dados do arquivo de teste resultou em um acerto de apenas 33.33% (contra 50% de antes). A matriz de confusão reflete essa piora no modelo, quando a compararmos com a do modelo antigo:**

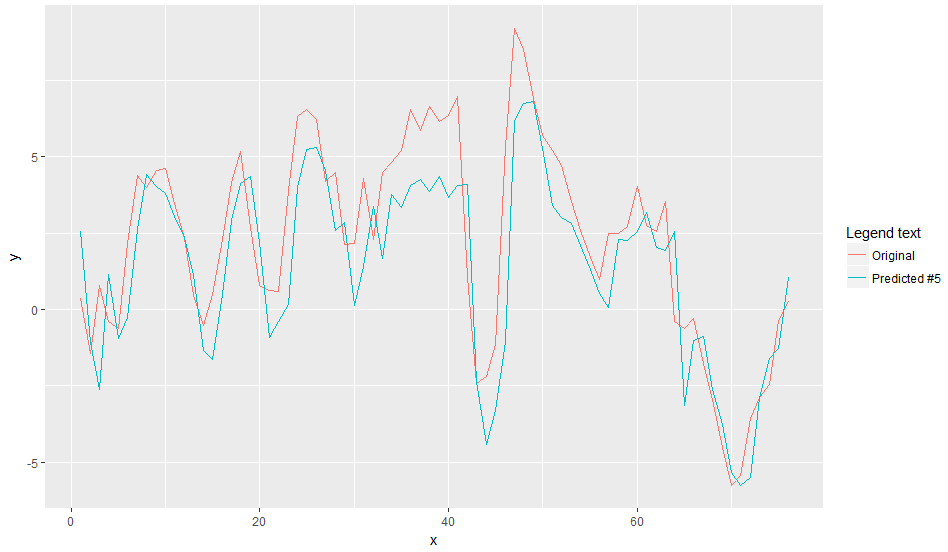


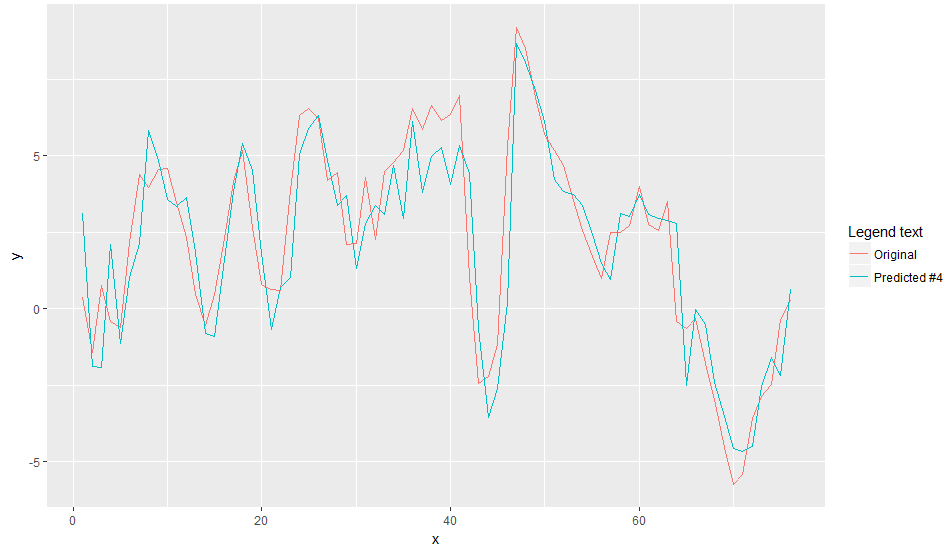
**Modelo antigo Modelo novo**

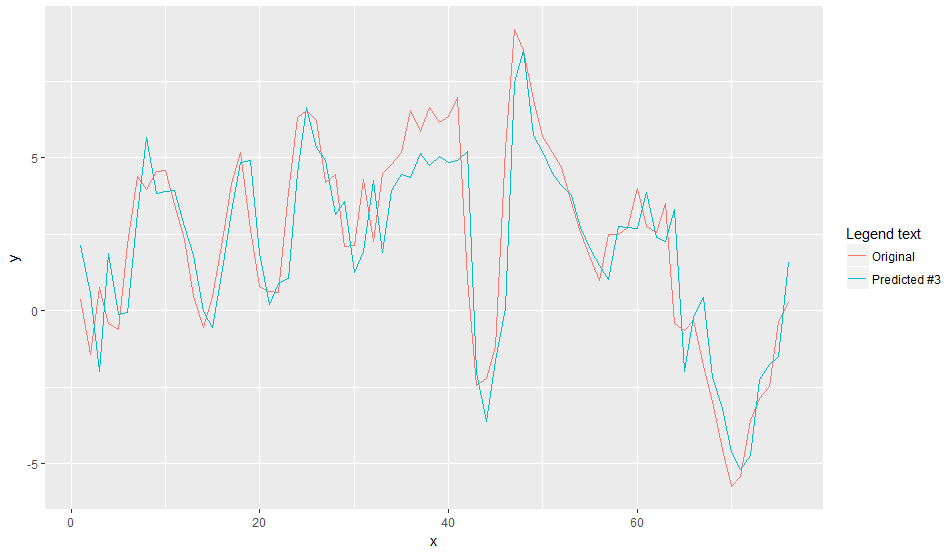
1. Utilizando o conjunto de dados PIBBrasil.csv e o pacote “RSNNS” do R, monte uma rede neural de Elman para prever o valor do PIB um passo à frente utilizando uma janela deslizante de tamanho *n*. O valor de *n* deverá ser avaliado entre os valores de 3 a 6. Varie também o número de neurônios na camada escondida. Utilize os 4 últimos valores do PIB para o conjunto de teste e os demais para o conjunto de treinamento.
2. Como resultado, preencha a tabela abaixo com as cinco configurações (*n* versus número de neurônios na camada escondida) com menor valor de RMSE para o conjunto de teste.

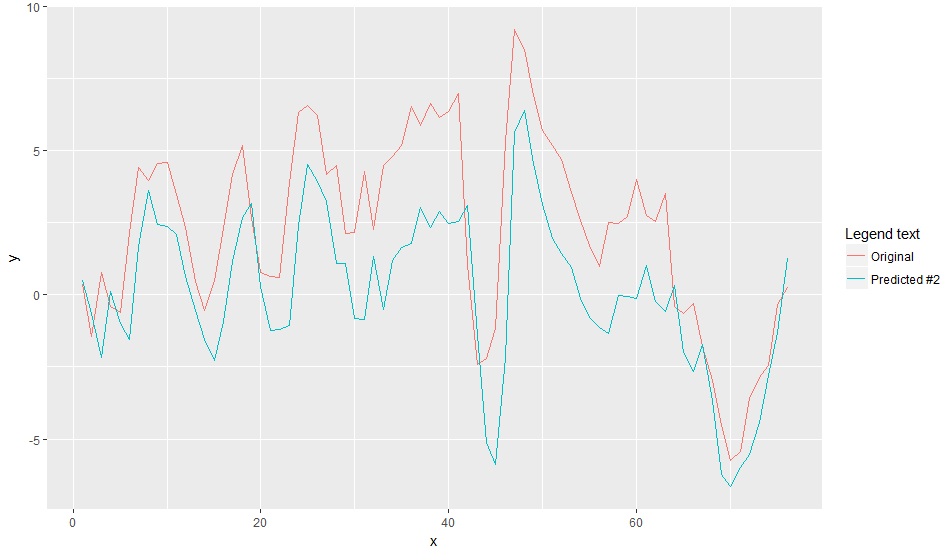
|  |  |  |  |
| --- | --- | --- | --- |
| **Janela** | **Número de Neurônios** | **RMSE**  **de Treinamento** | **RMSE**  **de Teste** |
| 3 | 6 | 1.556645 | 1.040738 |
| 3 | 5 | 1.500314 | 1.052697 |
| 3 | 4 | 1.543523 | 1.091918 |
| 4 | 4 | 1.756668 | 1.093390 |
| 3 | 13 | 1.574219 | 1.096785 |

1. Plote um gráfico com a série original do PIB e com o resultado da previsão para as cinco configurações escolhidas.











1. Você vai passar um mês em uma região selvagem e estará levando uma mochila, no entanto, o peso máximo que pode carregar é de 20kg. Você tem uma série de itens de sobrevivência disponíveis, cada um com seu próprio número de "pontos de sobrevivência", conforme tabela abaixo. Seu objetivo é maximizar o número de pontos de sobrevivência. Resolva o problema apresentando a lista de itens a serem selecionados usando Algoritmo Genético (pacote “genalg”) no ambiente R.

|  |  |  |
| --- | --- | --- |
| **Item** | **Ponto de sobrevivência** | **Peso** |
| A | 10 | 1 |
| B | 20 | 5 |
| C | 15 | 10 |
| D | 2 | 1 |
| E | 30 | 7 |
| F | 10 | 5 |
| G | 30 | 1 |



**Melhor solução encontrada: B+E+F+G (score = 90; peso = 18)**

1. Implementar um SIF no Matlab usando o Fuzzy Logic Toolbox para controlar um chuveiro. As variáveis de entrada são a temperatura da água e o fluxo de água. As variáveis de saída são duas, a taxa na qual as válvulas de água fria e de água quente devem ser abertas ou fechadas. Os conjuntos fuzzy de entrada e saída bem como as regras estão apresentadas abaixo. Após implementado, avalie as saídas das válvulas para os pares de entrada [Temperatura Fluxo] = [-12 -0.6; -5 0.2; 0 0; 5 0.2; 12 0.6]. Deverá ser entregue o arquivo “fis” do Matlab com o nome chuveiro.fis e o resultado das simulações.

|  |  |
| --- | --- |
|  |  |
|  |  |

If (temp is cold) and (flow is soft) then (cold is openSlow)(hot is openFast) (1)

If (temp is cold) and (flow is good) then (cold is closeSlow)(hot is openSlow) (1)

If (temp is cold) and (flow is hard) then (cold is closeFast)(hot is closeSlow) (1)

If (temp is good) and (flow is soft) then (cold is openSlow)(hot is openSlow) (1)

If (temp is good) and (flow is good) then (cold is steady)(hot is steady) (1)

If (temp is good) and (flow is hard) then (cold is closeSlow)(hot is closeSlow) (1)

If (temp is hot) and (flow is soft) then (cold is openFast)(hot is openSlow) (1)

If (temp is hot) and (flow is good) then (cold is openSlow)(hot is closeSlow) (1)

If (temp is hot) and (flow is hard) then (cold is closeSlow)(hot is closeFast) (1)

**Resultado:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Temp** | **Flow** | **Cold** | **Hot** |
| **-12** | **-0.6** | **0.3000** | **0.6350** |
| **-5** | **0.2** | **0.2689** | **0.0228** |
| **0** | **0** | **0.0000** | **0.0000** |
| **5** | **0.2** | **0.0228** | **0.2689** |
| **12** | **0.6** | **0.3000** | **0.6350** |