

**Projeto WEKA**

**23/07/2021**

Objetivo: Estudo das tarefas de agrupamento e classificação de dados empregando o software WEKA – **modelo K-médias e rede neural MLP.**

Etapas:

1. Descrição do software WEKA (disponível em https://[www.cs.waikato.ac.nz/ml/weka/).](http://www.cs.waikato.ac.nz/ml/weka/))

Weka é Software de aprendizado de máquina em Java de código aberto emitido sob a [GNU General Public License](http://www.gnu.org/licenses/gpl.html). Foi desenvolvido pela Universidade de Waikato na Nova Zelândia. Sua sigla significa Waikato Environment for Knowledge Analysis (Ambiente de Análise de Conhecimento de Waikato). Apresenta uma coleção de algoritmos de aprendizado de máquina para tarefas de mineração de dados. Ele contém ferramentas para preparação de dados, classificação, regressão, clustering, mineração de regras de associação e visualização.

1. Descrição dos conjuntos de dados:
   1. “Iris Plant database” (Disponível em <http://www.hakank.org/weka/iris.arff>)

O Iris Plant Database é um base de dados de reconhecimento de padrões. O conjunto de dados contém 3 classes com 50 instâncias cada, onde as classes referenciam a um tipo de planta íris. O atributo dessa base de dados é composto de:

* *Sepal length in cm* (comprimento das sépalas em centímetros)
* *Sepal width in cm* (largura das sépalas em centímetros)
* *Petal length in cm* (comprimento das pétalas em centímetros)
* *Petal width in cm* (largura das pétalas em centímetros)



Figura 01 – Alguns tipos de planta com Flor Iris

A base de dados, portanto, contém um estudo realizado sobre as plantas com flor do tipo Iris *(Figura 01).* Ele realiza a classificação dessas plantas de acordo com o tamanho da largura e comprimento de suas pétalas e sépalas. Elas podem ser classificadas *(Figura 02)* em Iris:

* Setosa,
* Versicolour,
* Virginica.

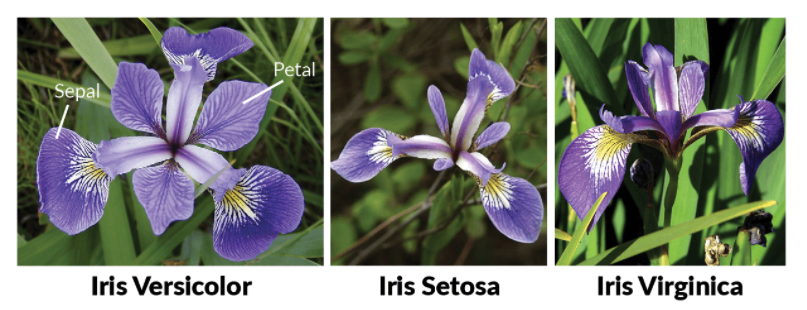


Figura 02 – Classificações da planta Iris

* 1. “Wine recognition data” (Disponível em <http://www.hakank.org/weka/wine.arff>)

Esta base de dados é resultado de um estudo químico realizado em vinhos produzido numa mesma região da Itália, mas que originaram de três cultivos diferentes. Com as características dos vinhos é possível determinar sua origem.

A análise determinou a quantidade de 13 elementos encontrados em cada um dos três tipos de vinhos. Os 13 atributos são:

1) Álcool  
2) Ácido málico  
3) Cinzas  
4) Alcalinidade das cinzas  
5) Magnésio  
6) Fenóis totais  
7) Flavanóides  
8) Fenóis não flavonóides  
9) Proantocianinas  
10) Intensidade de cor  
11) Matiz  
12) OD280 / OD315 de vinhos diluídos  
13) Prolina

Na base de dados Wine temos 3 classes com os seguintes número de instâncias:

* Classe 1: 59 instâncias;
* Classe 2: 71 instâncias;
* Classe 3: 48 instâncias.

Todos os atributos são contínuos, mas recomenda-se uma padronização das variáveis para certas aplicações. Nos teste desse trabalho serão usadas 13 variáveis reais e uma que terá o número da classe.

1. Descrição do modelo K-médias implementado no WEKA através da função de agrupamento “SimpleKMeans”.
2. Detalhar a função dos parâmetros disponíveis para configuração do modelo.

* **seed** - Define o valor inicial do ponto de partida (centro) com um número aleatório
* Paramento utilizado na função: Valor inicial.
* **displayStdDevs** - Exibe desvios padrão de atributos numéricos e contagens de atributos nominais. A função define o desvio padrão e a contagem nominal. Deve ser exibido na saída de cluster.
  + Parâmetro utilizado na função: true para exibir e false para não exibir.
* **numExecutionSlots** - O número de slots de execução (threads) a serem usados. Definir igual ao número de cpu / núcleos disponíveis. Ou seja a função define o grau de paralelismo a ser utilisado.
* Parâmetro utilizado na função: o número de tarefas a serem executadas em paralelo.
* **canopyMinimumCanopyDensity** - Se usar agrupamento de dossel para inicialização e / ou aumento de velocidade, esta é a densidade baseada em T2 mínima abaixo da qual um dossel será podado durante a poda periódica. A função denine a densidade mínima de base T2 abaixo da qual um dossel será podado durante a poda periódica.
* Parâmetro utilizado na função: a densidade mínima do dossel.
* **dontReplaceMissingValue**s ​​– Para Substituir os valores ausentes globalmente por média / modo. A função define se os valores em falta devem ser substituídos.
* Parâmetro: recebe true se os valores em falta forem substituídos e false se os valores em falta não forem substituídos.
* **debug** – A função envia informaçoes adficionais apara o console.
* Parâmetro utilizado na função: Se definido como True o clusterer pode enviar informações adicionais para o console se false não sera enviado informaçoes adicionais.
* **canopyT2** - A distância T2 a ser usada ao usar agrupamento de dossel. Valores < 0 indicam que isso deve ser definido usando uma heurística baseada no desvio padrão do atributo. A função define o raio t2 para usar quando o agrupamento de copa está sendo usado como pontos de partida e / ou para reduzir o número de distância
* Parâmetro utilizado na função: o raio t2 (distância).
* **numClusters** - Define o número de clusters.
* Parâmetro utilizado na função: o número de clusters.
* **doNotCheckCapabilities** – Realiza ou nao a verificação dos recursos do clusterer antes que o clusterer seja construído. Utilise com cuidado para reduzir o tempo de execução.
* Parâmetro utilizado na função: Se definido True, os recursos do clusterer não são verificados antes que o clusterer seja construído. Se false São verificadosantes da construção.
* **maxIterations** - Define o número máximo de iterações a serem executadas.
* Parâmetro utilizado na função: o número máximo de iterações
* **preserveInstancesOrder** - Preserva a ordem das instâncias. A função define se a ordem das instâncias deve ser preservada ou alterada.
* Parâmetro utilizado na função: true se os valores em falta forem substituídos e false para não alterar.
* **canopyPeriodicPruningRate** - Se estiver usando agrupamento de dossel para inicialização e / ou aumento de velocidade, esta é a frequência com que podar os dosséis de baixa densidade durante o treinamento. A função, entao, define a frequência para podar os copos de baixa densidade durante o treinamento.
* Parâmetro utilizado na função: a frequência (todas as instâncias) para podar depósitos de baixa densidade.
* **canopyMaxNumCanopiesToHoldInMemory** - Se estiver usando o clustering de canopy para inicialização e / ou aumento de velocidade, esse é o número máximo de candidatos a serem retidos na memória principal durante o treinamento do clusterer de canopy. A funação, portanto, define o número máximo de copos candidatos para reter na memória durante o treinamento. A distância T2 e as características dos dados determinam a quantidade de copos candidatos que se formam antes da poda periódica e final serem realizadas. Pode não haver memória suficiente disponível se o T2 estiver definido muito baixo.
* Parâmetro utilizado na função: o número máximo de copos candidatos para reter na memória durante o treino
* **initializationMethod** - Defina o método de inicialização a ser usado, aleatório, k-means ++, Canopy ou mais distante primeiro.
* Parâmetro utilizado na função: o método de inicialização a ser utilizado.
* **distanceFunction** – Define a função de distância a ser usada para comparação de instâncias..
* Parâmetro utilizado na função: uma função de distância
* **canopyT1** - A distância T1 a ser usada ao usar agrupamento de dossel. Valores < 0 são tomados como um multiplicador positivo para a distância T2. A função, entao, define o raio T1 para ser utilizado quando o agrupamento de copa está sendo utilisado como pontos de partida e / ou para reduzir o número de distância.
* Parâmetro utilizado na função: o raio T1 (distância)
* **fastDistanceCalc** - Usa valores de corte para acelerar o cálculo da distância, mas suprime também o cálculo e a saída da soma dos erros quadrados / soma das distâncias dentro do cluster. A função entao define se é necessário usar um cálculo de distância mais rápido.
* Parâmetro utilizado na função: true se um cálculo mais rápido for utilisar e false se não for utilizar
* **reduzNumberOfDistanceCalcsViaCanopies** - Usa o agrupamento de canopy para reduzir o número de cálculos de distância realizados por k-means. A função, portanto, define se os canopies podem ser utilisados para reduzir o número de cálculos de distância necessários.
* Parâmetro utilizado na função: True se os copos devem ser utilisados para reduzir o número de cálculos de distância e false para não utilizar os copos.

1. Identificar os parâmetros referentes ao número de grupos, número máximo de iterações e função de distância.

* Parametro referente ao número de grupos:
* Função: numClusters
* Parâmetro: O número de clusters.
* Parametro referente ao número máximo de iterações:
* Funçao: MaxIterations
* Parâmetro: O número máximo de iterações
* Parametro referente à função de distância:
* Funçao: DistanceFunction
* Parâmetro: Função de distância

1. Executar os testes abaixo para os dois conjuntos de dados mencionados no item 2.
2. Experimento 1: identificar o número de classes existentes no conjunto de dados e empregar o mesmo valor como número de grupos do modelo k-médias.

* **Iris Plant database:**
* Possui 3 classes: Iris-setosa, Iris-versicolor, Iris-virginica;
* **numClusters = 03;**
* Usando o modelo K-médias e empregando o mesmo valor do nºde classes como número de grupos obtêm os seguintes grupos com as seguintes porcentagens:

0 - (33%) com 50 instancias agrupadas – Classe Iris-setosa

1 - (33%) com 50 instancias agrupadas – Classe Iris-versicolor

2 - (33%) com 50 instancias agrupadas – Classe Iris-virginica

* **Wine recognition data:**
* Possui 03 classes: Três cultivos diferentes (classe 1, classe 2 e classe 3)
* **numClusters = 03;**
* Utilizando o modelo K-médias e empregando o mesmo valor como número de grupos obtemos os seguintes grupos com as seguintes porcentagens:

0 - (33%) com 59 instancias agrupadas – Classe 1

1 - (27%) com 48 instancias agrupadas – Classe 3

2 - (40%) com 71 instancias agrupadas – Classe 2

1. Experimento 2: modificação do número de grupos – valores: {2, 3, 4, e 5}.

* **Iris Plant database:** valores do numClusters: {2, 3, 4, e 5}

Percebe-se que ao aumentar o número de grupos a soma dos erros quadráticos do cluster diminui, porém uma mesma classe pode ficar reagrupada em dois grupos, sendo:

* **Valor numClusters = 2**
* Within cluster sum of squared errors: 62.14368828157970
* Com duas classes: classe Iris-versicolor com 100 instâncias (67%) e classe Iris- setosar com 50 instancias (33%)
* **Valor numClusters = 3**
* Within cluster sum of squared errors: 7.817456892309574
* Com três classes: classe Iris-versicolor com 50 instâncias (33%), classe Iris- setosa com 50 instancias (33%) e classe Iris-virginica com 50 instancias (33%).
* **Valor numClusters = 4**
* Within cluster sum of squared errors: 6.613823274690356
* Com três classes: duas classes, sendo classe Iris-versicolor com 24 instâncias (16%) e com 26 intancias (17%), classe Iris-setosa com 50 instancias (33%) e classe Iris-virginica com 50 instancias (33%).
* **Valor numClusters = 5**
* Within cluster sum of squared errors: 6.293556861810846
* Com três classes: três classes sendo, duas classes Iris-versicolor com 19 instâncias (13%) e uma com 12 intancias (8%), classe Iris-setosa com 50 instancias (33%) e classe Iris-virginica com 50 instancias (33%).
* **Wine recognition data** valores do numClusters: {2, 3, 4, e 5}

Percebe-se que ao aumentar o número de grupos a soma dos erros quadráticos do cluster diminui, porém uma mesma classe ficar reagrupada em dois grupos, sendo:

* + **Valor numClusters = 2**
* Within cluster sum of squared errors: 118.21713874953554
* Com 2 classes: classe 1 com 59 instancias (33%) e classe 2 com 119 instancias (67%).
  + **Valor numClusters = 3**
* Within cluster sum of squared errors: 49.998510705570595
* Com 3 classes: classe 1 com 59 instancias (33%), classe 2 com 71 instancias (40%) e e classe 3 com 48 instancias (27%).
  + **Valor numClusters = 4**
* Within cluster sum of squared errors: 45.196536193467495
* Com 3 classes: classe 1 com 59 instancias (33%) ,duas classes 2, uma com 37 instancias (21%) e outra com 34 instancias (19%) e classe 3 com 48 insatancias (27%).
  + **Valor numClusters = 5**
  + Within cluster sum of squared errors: 43.2346225206482
  + Com 3 classes: Duas classe 1, uma com 30 instancias (17%) e uma com 29 instancias (16%), duas classes 2, uma com 37 instancias (21%) e outra com 34 instancias (19%) e classe 3 com 48 insatancias (27%).
  1. Montar novos experimentos variando a semente (parâmetro “seed”) com o objetivo de avaliar a variação no resultado obtido a partir de centros iniciais diferentes.
* **Iris Plant database**

Utilizando 3 classes e os seguintes valores para o parâmetro seed {2,5,10,20} para avaliar a variação no resultado obtido através dos pontos de partida iniciais aleatórios.

seed = 2

Cluster 0: 6.6,2.9,4.6,1.3,**Iris-versicolor**

Cluster 1: 5.5,2.3,4,1.3,**Iris-versicolor**

Cluster 2: 5.1,3.8,1.9,0.4,**Iris-setosa**

seed = 5

Cluster 0: 6.4,3.1,5.5,1.8,**Iris-virginica**

Cluster 1: 5.6,2.7,4.2,1.3,**Iris-versicolor**

Cluster 2: 6.7,3.1,4.7,1.5,**Iris-versicolor**

seed = 10

Cluster 0: 6.1,2.9,4.7,1.4,**Iris-versicolor**

Cluster 1: 6.2,2.9,4.3,1.3,**Iris-versicolor**

Cluster 2: 6.9,3.1,5.1,2.3,**Iris-virginica**

seed = 15

Cluster 0: 6.9,3.1,5.1,2.3,**Iris-virginica**

Cluster 1: 7.7,2.8,6.7,2,**Iris-virginica**

Cluster 2: 4.6,3.6,1,0.2,**Iris-setosa**

seed = 20

Cluster 0: 5.5,2.3,4,1.3,**Iris-versicolor**

Cluster 1: 5.8,2.7,5.1,1.9,**Iris-virginica**

Cluster 2: 4.9,3.1,1.5,0.1,**Iris-setosa**

Os 3 pontos iniciais de partida variam conforme o numero do seed fornecido. Estes podem inicialmente ser definidos como sendo do mesmo grupo, mas obviamente são reagrupadas apos a definição via os calculos da distância nas proximas interações.

* **Wine recognition data**

Utilizando 3 classes e os seguintes valores para o parâmetro seed {2,5,10,15,20} para avaliar a variação no resultado obtido através dos pontos de partida iniciais aleatórios.

seed = 2

Cluster 0: 13.51,1.8,2.65,19,110,2.35,2.53,0.29,1.54,4.2,1.1,2.87,1095,**1**

Cluster 1: 13.49,3.59,2.19,19.5,88,1.62,0.48,0.58,0.88,5.7,0.81,1.82,580,**3**

Cluster 2: 13.11,1.9,2.75,25.5,116,2.2,1.28,0.26,1.56,7.1,0.61,1.33,425,**3**

Seed = 5

Cluster 0: 13.69,3.26,2.54,20,107,1.83,0.56,0.5,0.8,5.88,0.96,1.82,680**,3**

Cluster 1: 12.45,3.03,2.64,27,97,1.9,0.58,0.63,1.14,7.5,0.67,1.73,880,**3**

Cluster 2: 12.86,1.35,2.32,18,122,1.51,1.25,0.21,0.94,4.1,0.76,1.29,630,**3**

Seed = 10

Cluster 0: 13.3,1.72,2.14,17,94,2.4,2.19,0.27,1.35,3.95,1.02,2.77,1285,**1**

Cluster 1: 12.22,1.29,1.94,19,92,2.36,2.04,0.39,2.08,2.7,0.86,3.02,312,**2**

Cluster 2: 11.61,1.35,2.7,20,94,2.74,2.92,0.29,2.49,2.65,0.96,3.26,680,**2**

Seed = 15

Cluster 0: 12.37,1.17,1.92,19.6,78,2.11,2,0.27,1.04,4.68,1.12,3.48,510,**2**

Cluster 1: 13.05,1.77,2.1,17,107,3,3,0.28,2.03,5.04,0.88,3.35,885,**1**

Cluster 2: 13.11,1.01,1.7,15,78,2.98,3.18,0.26,2.28,5.3,1.12,3.18,502,**2**

Seed = 20

Cluster 0: 12.7,3.55,2.36,21.5,106,1.7,1.2,0.17,0.84,5,0.78,1.29,600,**3**

Cluster 1: 12.67,0.98,2.24,18,99,2.2,1.94,0.3,1.46,2.62,1.23,3.16,450,**2**

Cluster 2: 12.07,2.16,2.17,21,85,2.6,2.65,0.37,1.35,2.76,0.86,3.28,378,**2**

Seed = 25

Cluster 0: 13.71,5.65,2.45,20.5,95,1.68,0.61,0.52,1.06,7.7,0.64,1.74,740**,3**

Cluster 1: 12.93,3.8,2.65,18.6,102,2.41,2.41,0.25,1.98,4.5,1.03,3.52,770,**1**

Cluster 2: 13.05,3.86,2.32,22.5,85,1.65,1.59,0.61,1.62,4.8,0.84,2.01,515,**2**

Os 3 pontos iniciais de partida da wine também variam conforme o numero do seed fornecido. Estes podem inicialmente ser definidos como sendo do mesmo grupo, mas obviamente são reagrupados apos a definição via os calculos da distância nas proximas interações.

Entao:

Apesar de o centro incial ser sempre alterado por seed, o centroid será alterado apos cada interação, pela media dos pontos de cada um dos grupos. Sendo o centroid definido por tres pontos no Wine e na Ires (um ponto centroid para cada grupo) e estes mesmos pontos podem inicialmente pertecerem a um determinado grupo, mas no decorrer das interações podem passar a pentecer a qualquer um dos demais grupos, conforme assim determinar o calculo da distancia de cada um dos pontos (Função Euclidean distance) ate o centroide e tambem da media que determinará o novo centroide da proxima interação, assim o resultado final dos grupos não mudaram conforme o Seed foi sendo alterado, mas os pontos inciais escolhidos num primeiro momento podem pentencer a uma mesmo grupo e no decorrer da execução e interações os mesmos serão reagrupados em grupos iguais ou não, uma vez que a distância dos pontos ate os novos centroides definidos no decorrer das interações pode ser menor do que a distância dada pelos pontos ate os centroides anteriores (iniciais), assim ocorrerá a recolocação dos centroides iniciais aos grupos conforme a menor distância.

1. Descrição do modelo MLP (Multi Layer Perceptron) implementado no WEKA através da função de classificação “MultilayerPerceptron”.
2. Detalhar a função dos parâmetros disponíveis para configuração do modelo.

**Seed -** Inicializará o gerador de número aleatório. Números aleatórios são usados ​​para definir os pesos iniciais das conexões entre os nós e também para embaralhar os dados de treinamento.

* Parametro da função: numero do valor do Peso inicial

**momentum** - Aplicará às atualizações de peso.

* Parametro da função: Taxa do Peso a atualizar

**nominalToBinaryFilter** - Isso irá pré-processar as instâncias com o filtro NominalToBinary. Isso pode ajudar a melhorar o desempenho se houver atributos nominais nos dados.

* Parametro da função: True filtra os atributos nominais e false não realiza filtragem.

**hiddenLayers** – Definirá as camadas ocultas da rede neural. Esta é uma lista de números inteiros positivos, separados por vírgulas.

* Parametro da função: 1 para cada camada oculta. Para não ter camadas ocultas, coloque um único 0. Esta definição de camada só será usada se a construção automática estiver definida. Também existem valores curinga:
  + 'a' = (atributos + classes) / 2,
  + 'i' = atributos,
  + 'o' = classes,
  + 't' = atributos + classes.

**validationThreshold** - encerrará o teste de validação. O valor dita quantas vezes consecutivas o erro do conjunto de validação pode piorar antes que o treinamento seja encerrado.

* Parametro da função: numero de vezes de erro consecutivo para encerrar

**GUI -** abre uma interface gui. Isso permitirá a pausa e alteração da rede neural durante o treinamento.

**normalizeAttributes** – realizará a normalização dos atributos. Isso pode ajudar a melhorar o desempenho da rede. Isso não depende de a classe ser numérica. Isso também normalizará os atributos nominais (depois de serem executados através do filtro nominal para binário, se estiver em uso) para que os valores binários fiquem entre -1 e 1.

* Parametro da função: True para nomalizar e false para não nomralizar o valores.

**numDecimalPlaces** – Define o número de casas decimais a serem usadas para a saída de números no modelo.

* Parametro da função: o número de casas decimais

**batchSize** – define o número preferencial de instâncias a serem processadas se a predição em lote estiver sendo realizada. Mais ou menos instâncias podem ser fornecidas, mas isso dá às implementações a chance de especificar um tamanho de lote preferido.

* Parametro da função: numero de instancias

**decay** - Fará com que a taxa de aprendizagem diminua. Isso dividirá a taxa de aprendizado inicial pelo número de ciclos para determinar qual deve ser a taxa de aprendizado atual. Isso pode ajudar a impedir que a rede se desvie da saída alvo, bem como melhorar o desempenho geral. Observe que a taxa de aprendizado decrescente não será mostrada na GUI, apenas a taxa de aprendizado original. Se a taxa de aprendizado for alterada na GUI, isso será tratado como a taxa de aprendizado inicial.

* Parametro da função: true para diminuir a taxa de aprendizado e false para deixar constante

**validationSetSize** – determina que o treinamento continuará até que seja observado que o erro no conjunto de validação tem piorado consistentemente ou se o tempo de treinamento for atingido. Se for definido como zero, nenhum conjunto de validação será usado e, em vez disso, a rede treinará para o número especificado de ciclos.

* Parametro da função: O tamanho percentual do conjunto de validação

**trainingTime** – Define o número maximo de ciclos para treinar. Se o conjunto de validação for diferente de zero, ele pode encerrar a rede mais cedo.

* Parametro da função: número maximo de ciclos

**Debug -** A função envia informaçoes adicionais apara o console.

* Parâmetro da função: Se definido como True o classificador pode enviar informações adicionais para o console se false não sera enviado informaçoes adicionais.

**resume** - Definirá se o classificador pode continuar o treinamento após realizar o número solicitado de iterações. Observe que definir isso como verdadeiro irá reter certas estruturas de dados que podem aumentar o tamanho do modelo.

* Parâmetro da função: Se definido como True o classificador poderá continua se false sera encerrado apos o fim do limite Maximo das iterações.

**autoBuild** – realiza a adicão e conecxão das camadas ocultas na rede.

* Parâmetro da função: Se definido como True realiza a adição e conecção da rede automaticamente se false não realiza a adiçãoe conecxão.

**normalizeNumericClass** – Realiza a nomalização da classe se for numérica. Isso pode ajudar a melhorar o desempenho da rede. Ele normaliza a classe para estar entre -1 e 1. Observe que isso ocorre apenas internamente, a saída será reduzida para a faixa original.

* Parametro da função: True para nomalizar e false para não normalizar os valores.

**learningRate** – define a taxa de aprendizagem para atualizações de peso.

* Parametro da função: Taxa de aprendizagem

**doNotCheckCapabilities** - Realiza ou nao a verificação dos recursos do classificador antes que o mesmo seja construído. Utilise com cuidado para reduzir o tempo de execução.

* Parâmetro utilizado na função: Se definido True, os recursos do classificador não são verificados antes que o classificador seja construído. Se false São verificados antes da construção.

**reset** - Permitirá que a rede seja redefinida com uma taxa de aprendizado mais baixa. Se a rede divergir da resposta, isso redefinirá automaticamente a rede com uma taxa de aprendizado mais baixa e começará o treinamento novamente. Esta opção está disponível apenas se a GUI não estiver configurada. Observe que, se a rede diverge, mas não é permitida a redefinição, ela falhará no processo de treinamento e retornará uma mensagem de erro.

* Parâmetro utilizado na função: True para permitir diminuir a taxa de aprendizado e false para não permitir a redefinição da taxa

1. Identificar os parâmetros referentes à taxa de aprendizado, momento e número de épocas.

* Parametro referente à taxa de aprendizado:
* Função: learningRate
* Parâmetro: Taxa de aprendizagem
* Parametro referente ao momento:
* Função: momentum
* Parâmetro: Taxa do Peso
* Parametro referente ao numero de épocas:
* Função: trainingTime
* Parâmetro: Número máximo de ciclos

1. Executar os testes abaixo para os dois conjuntos de dados mencionados no item 2. Para estas execuções deve ser selecionada a opção de teste **“Cross-validation”, com 10 pastas** conforme padrão do WEKA. Esta opção implementa a validação cruzada.

* **Iris Plant database:**

1. Experimento 1: modificação da taxa de aprendizado – valores: {0,3; 0,6; 0,9}

Momento: 0,2 e Época: 500 (Fixos)

1. **Taxa de Aprendizado: 0,3**

* A taxa de acerto foi de 97,33%, com 146 intancias.
* A taxa de erro foi de 2,66%, com 4 instancias, sendo que duas Versicolors foram classificadas como Virgínica e duas Virgínicas foram classificadas como Versicolor.

1. **Taxa de Aprendizado: 0,6**

* As mesmas taxas e resultados quando executado com Taxa de Aprendizado = 0,3 (Os Mesmos resultdo obtidos em a) I ).

1. **Taxa de Aprendizado: 0,9**

* A taxa de acerto foi de 96,67% com 145 instancias
* A taxa de erro foi de 3,33%, com 5 instancias, sendo que três Versicolors foram classificadas como Virgínica e duas Virgínicas foram classificadas como Versicolor

1. Experimento 2: modificação do momento – valores: {0,2 0,5; 0,8}

Taxa de Aprendizado: 0,3 e Época: 500 (Fixos)

1. **Momento: 0,2**

* Executado em a) I
* Mesmos resultado obtido em a) I e II.

1. **Momento: 0,5**

* A taxa de acerto foi de 97,33%, com 146 instancias.
* A taxa de erro foi de 2,66 %, com 4 instancias, sendo que duas Versicolors foram classificadas como Virgínica e duas Virgínicas foram classificadas como Versicolor.
* Mesmos resultdos obtidos em a) I, II e b) I

1. **Momento: 0,8**

* A taxa de acerto foi de 97,33%,
* A taxa de erro foi de 2,66 %, com 4 instancias, sendo que duas Versicolors foram classificadas como Virgínica e duas Virgínicas foram classificadas como Versicolor.
* Mesmos resultdos obtidos em a) I e II e b) I e II

1. Experimento 3: modificação do número de épocas: {500, 1000, 2000}

Taxa de Aprendizado: 0,3; Momento: 0,2 (Fixos)

1. **Época: 500**

* Executado em a) I
* Mesmos resultados obtido em a) I e II e b) I, II e III

1. **Época: 1000**

* A taxa de acerto foi de 97,33%,
* A taxa de erro foi de 2,66 %, com 4 instancias, sendo que duas Versicolors foram classificadas como Virgínica e duas Virgínicas foram classificadas como Versicolor.
* Mesmos resultados obtido em a) I e II e b) I , II e III e c) I

1. **Época: 2000:**

* A taxa de acerto foi de 96,67% com 145 instancias
* A taxa de erro foi de 3,33%, com 5 instancias, sendo que três Versicolors foram classificadas como Virgínica e duas Virgínicas foram classificadas como Versicolor
* O Mesms resultado obtido em a) III

1. Montar novos experimentos combinando taxa de aprendizado e momento. Os valores padrão do WEKA são 0,3 e 0,2, respectivamente. Defina alguns experimentos combinando os valores propostos nos itens a) e b). Não é necessário fazer todas as combinações.

**Época: 2000 (Fixo)**

1. **Taxa de Aprendizado: 0,5; Momento: 0,1**

* A taxa de acerto foi de 96,67% com 145 instancias
* A taxa de erro foi de 3,33%, com 5 instancias, sendo que três Versicolors foram classificadas como Virgínica e duas Virgínicas foram classificadas como Versicolor
* O Mesmo resultado obtido em a) III e c) III

1. **Taxa de Aprendizado: 0,1; Momento: 0,5**

* A taxa de acerto foi de 96,67% com 145 instancias
* A taxa de erro foi de 3,33%, com 5 instancias sendo que quatro Versicolors foram classificadas como Virgínica e uma Virgínica foi classificada como Versicolor.
* Os Mesmos resultado obtido em a) III e c) III d) I , mas com diferença na classificação errada. Uma versicolor a mais classificada erroneamente.
* **Wine recognition data:**

1. Experimento 1: modificação da taxa de aprendizado – valores: {0,3; 0,6; 0,9}

Momento: 0,2; Época: 500 (Fixos)

1. **Taxa de Aprendizado: 0,3**

* A taxa de acerto foi de 97,191% em 173 instancias;
* A taxa de erro foi de 2,809 em 5 instancias, sendo que dois vinhos da classe 2 foram classificados como classe 1 e dois como classe 3 e ainda um vinho da classe 3 foi classificado como da classe 2.

1. **Taxa de Aprendizado: 0,6**

* Taxa de acerto foi de 97,191% em 173 instancias;
* A taxa de erro foi de 2,809 em 5 instancias, sendo que dois vinhos da classe 2 foram classificados como classe 1 e dois como classe 3 e um vinho da classe 3 foi classificado como classe 2.
* O Mesmo resultado obtido em a) I

1. **Taxa de Aprendizado: 0,9**

* A taxa de acerto foi de 97,191% em 173 instancias;
* A taxa de erro foi de 2,809 em 5 instancias, sendo que dois vinhos da classe 2 foram classificados como classe 1 e dois como classe 3 e um vinho da classe 3 foi classificado como classe 2.
* O Mesmo resultado obtido em a) I e II

1. Experimento 2: modificação do momento – valores: {0,2 0,5; 0,8}

Taxa de Aprendizado: 0,3; Época: 500 (Fixos)

1. **Momento: 0,2**

* Executado em a) I
* Mesmos resultados obtido em a) I , II e III

1. **Momento: 0,5**

* A taxa de acerto foi de 97,191% em 173 instancias;
* A taxa de erro foi de 2,809 em 5 instancias, sendo que dois vinhos da classe 2 foram classificados como classe 1 e dois como classe 3 e um vinho da classe 3 foi classificado como classe 2.
* O Mesmo resultado obtido em a) I , II e II e b) I

1. **Momento: 0,8**

* A taxa de acerto foi de 97,191% em 173 instancias;
* A taxa de erro foi de 2,809 em 5 instancias, sendo que dois vinhos da classe 2 foram classificados como classe 1 e dois como classe 3 e um vinho da classe 3 foi classificado como classe 2.
* O Mesmo resultado obtido em a) I , II e II e b) I e II

1. Experimento 3: modificação do número de épocas: {500, 1000, 2000}

Taxa de Aprendizado: 0,3; Momento: 0,2 (Fixos)

1. **Época: 500**

* Executado em a) I
* Mesmos resultados obtido em a) I, II e III e b) I, II e III

1. **Época: 1000**

* A taxa de acerto foi de 97,191% em 173 instancias;
* A taxa de erro foi de 2,809 em 5 instancias, sendo que dois vinhos da classe 2 foram classificados como classe 1 e dois como classe 3 e um vinho da classe 3 foi classificado como classe 2.
* O Mesmo resultado obtido em a) I, II e II e b) I, II e III e c) I

1. **Época: 2000**

* A taxa de acerto foi de 97,191% em 173 instancias;
* A taxa de erro foi de 2,809 em 5 instancias, sendo que dois vinhos da classe 2 foram classificados como classe 1 e dois como classe 3 e um vinho da classe 3 foi classificado como classe 2.
* O Mesmo resultado obtido em a) I, II e II e b) I, II e III e c) I e II

1. Montar novos experimentos combinando taxa de aprendizado e momento. Os valores padrão do WEKA são 0,3 e 0,2, respectivamente. Defina alguns experimentos combinando os valores propostos nos itens a) e b). Não é necessário fazer todas as combinações.

Época: 2000:

* **Taxa de Aprendizado: 0,5; Momento: 0,1**
* A taxa de acerto foi de 97,191% em 173 instancias;
* A taxa de erro foi de 2,809 em 5 instancias, sendo que dois vinhos da classe 2 foram classificados como classe 1 e dois como classe 3 e um vinho da classe 3 foi classificado como classe 2.
* O Mesmo resultado obtido em a) I, II e II e b) I, II e III e c) I, II e III
* **Taxa de Aprendizado: 0,1; Momento: 0,5;**
* A taxa de acerto foi de 97,191% em 173 instancias;
* A taxa de erro foi de 2,809 em 5 instancias, sendo que dois vinhos da classe 2 foram classificados como classe 1 e dois como classe 3 e um vinho da classe 3 foi classificado como classe 2.
* O Mesmo resultado obtido em a) I, II e III e b) I, II e III e c) I, II e III e d) I

1. Avaliar os resultados obtidos na etapa 4)

Avaliando o resultado do modelo k-médias dos conjuntos de dados Iris Plant database e Wine recognition data percebe-se que ambos contém o mesmo numero de classes e ao aumentar o número de grupos à soma dos erros quadráticos do cluster diminui, porém quando se fornece um numero de cluster maior que o numero de classes o mesmo começa a apresentar os elementos de uma mesma classe em dois grupos diferentes, fazendo assim uma nova divisao da classe em mais de um grupo. E por fim, a respeito dos parâmetros “seed”, ao alterar o seu valor, o centroides tambem são alterados, mas o resultado final do agrupamento não sofre mudanças. Os pontos inciais escolhidos num primeiro momento podem pertencer a um mesmo grupo e no decorrer da execução e interações os mesmos serão reagrupados, uma vez que a distância dos pontos ate os novos centroides que são definidos no decorrer das interações podem ser menores do que a distancia euclediana dada pelos pontos ate os centroides anteriores (iniciais), assim pode ocorrer um novo reagrupamento dos pontos.

1. Avaliar os resultados obtidos na etapa 6), comentando cada resultado com relação à taxa de acerto do classificador. Para tanto, deve ser considerado o número de instâncias corretamente classificadas na saída produzida pelo WEKA.

* **Iris Plant database:**
* Para taxa de aprendizado 0,3 e 0,6
* Para momento: 0,2; 0,5 e 0,8
* Para Época: 500 e 1000
* **Nas experiências a) I e II e b) I , II e III e c) I e II.** Obteve:
* A taxa de acerto foi de 97,33%, **com 146 instancias.**
* A taxa de erro foi de 2,66%, com 4 instancias, sendo que duas Versicolors foram classificadas como Virgínica e duas Virgínicas foram classificadas como Versicolor.
* Para taxa de aprendizado 0,9 e
* Para Epoca 2000:
* **Nas experiências a) III e c) III d) I e II.** Obteve:
* A taxa de acerto foi de 96,67% **com 145 instancias**
* A taxa de erro foi de 3,33%, com 5 instancias, sendo que **três** Versicolors foram classificadas como Virgínica e **duas** Virgínicas foram classificadas como Versicolor
* A emperiência d) II teve a mesma taxa de erro foi de 3,33%, com 5 instancias , porém **quatro** Versicolors foram classificadas como Virgínica e **uma** Virgínica foi classificada como Versicolor.

Podemos inferir que o classificador MLP para o banco Iris Plant foi mais eficaz quando se optou por um numero menor de interações (epoca) e quando se aplicou uma taxa de aprendizado menor, uma vez que quando a taxa de aprendizado e o numero de epoca foi o mais alto entre as experiências realizadas, o resultado obtivo foi uma a taxa de erro um pouco maior e por isso o consequente aumento do numero de instancia errada.

* **Wine recognition data**
* Para taxa de aprendizado 0,3; 0,6 e 0,9
* Para momento: 0,2; 0,5 e 0,8
* Para Época: 500, 1000 e 2000
* **Nas experiências a) I, II e III e b) I, II e III e c) I, II e III e d) I e II.** Obteve:
* A taxa de acerto foi de 97,191% em **173 instancias;**
* A taxa de erro foi de 2,809 em 5 instancias, sendo que dois vinhos da classe 2 foram classificados como classe 1 e dois como classe 3 e um vinho da classe 3 foi classificado como classe 2.

Podemos inferir atravez das experiências realizadas que o classificador MLP para o banco Wine recognition foi indiferente em todas as experiencias realizadas, pois todas obtiveram a taxa de acerto de 97,191% com 173 intancias e taxa de erro de 2,809 com 5 instâncias. Todas apresentaram no total 5 erros. Destes erros 4 erros eram da classe 2, onde 2 foram classificados como sendo da classe 1 e 2 da classe 3 e 1 erro da classificação era da classe 3 que foi classificado pelo MLP como sendo da classe 2. Para Wine recognition o ideal seria utilizar uma taxa de aprendizado maior (para aprender mais rápido, ja que não há interferencia no erro quando a taxa de apredizado é maior) e optar por um numero de ciclo menor para ser mais eficiente (rápido), uma vez que, com ciclos maiores não interferiu, nestes casos executados, na taxa de erro ou acertos das classificaçoes dos 3 tipos de vinhos.

* **Conclusões do MLP para os bancos wine recognition e iris plant:**
* Ambos, tanto o Wine recognition e Ires Plant **foram indiferentes quanto ao momento** (Taxa do peso).
* Ambos, tanto o Wine recognition e Ires Plant devem optar por utilizar um **menor ciclo**, uma vez que o Wine recognition pode se tornar mais eficiente por se tornar mais rapido gastando menos tempo ao realizar menos interações e o irez Pant mostrou ser mais eficaz (com menor numero de erros)
* O **Ires Plant** deve apresenta **menor taxa de aprendizado** para ser mais eficaz.
* Enquanto o **Wine recognition** tem capacidade para **maior taxa de aprendizado**, uma vez que não obteve variação nos erros com taxa maiores, ao contrario do banco Ires Plant.

1. Faça alguns testes modificando o número de pastas da validação cruzada. Empregue um número menor de pastas, entre 2 e 5, por exemplo, mantendo os valores dos outros parâmetros da rede neural no padrão do WEKA.

* **Iris Plant database:**
* Taxa de Aprendizado: 0,3; Momento: 0,2; Época: 500 (Fixos)

1. Pastas: 2

* A taxa de acerto foi de 94,66% com 142 instancias
* A taxa de erro foi de 5,33% com 8 instancias,sendo que 6 Versicolors foram classificadas como Virgínica e 2 Virgínicas foram classificadas como Versicolors.

1. Pastas: 3

* A taxa de acerto foi de 94% com 142 instancias
* A taxa de erro foi de 6% com 9 instancias, sendo que 5 Versicolors foram classificadas como Virgínica e 4 Virgínicas foram classificadas como Versicolors.

1. Pastas: 4

* A taxa de acerto foi de 95,33% com 143 instancias
* A taxa de erro foi de 4,66% com 7 instancias, sendo que 4 Versicolors foram classificadas como Virgínica e 3 Virgínicas foram classificadas como Versicolors.

1. Pastas: 5

* A taxa de acerto foi de 96% com 144 instancias
* A taxa de erro foi de 4% com 6 intancias, sendo que 5 Versicolors foram classificadas como Virgínica e uma Virgínica foi classificada como Versicolor.
* **Wine recognition data**
* Taxa de Aprendizado: 0,3; Momento: 0,2; Época: 500 (Fixos)

1. Pastas: 2

* A taxa de acerto foi de 95,5% com 170 instancias
* A taxa de erro foi de 4,5% com 8 instancias,sendo que os 8 vinhos pertencem a classe 2 , mas foram classificadas erroneamente como 4 vinhos sendo da classe 1 e 4 vinhos sendo da classe 3.

1. Pastas: 3

* A taxa de acerto foi de 95,5% com 170 instancias
* A taxa de erro foi de 4,5% com 8 instancias,sendo que das 8 , cinco vinhos pertencem a classe 2 , mas foram classificados 2 como sendo classe 1 e 3 vinhos sendo classificados como classe 3 e as dois vinhos restantes são da classe 3, mas foram classificados como sendo classe 2.

1. Pastas: 4

* A taxa de acerto foi de 96,63% com 172 instancias
* A taxa de erro foi de 3,37% com 6 instancias,sendo que das 6 , cinco vinhos pertencem a classe 2 , mas foram classificadas como 1 vinho sendo classe 1 e 4 vinhos sendo da classe 3 e o ultimo vinho restante da classe 3 foi classificado erroneamente como sendo classe 2.

1. Pastas: 5 :

* A taxa de acerto foi de 97,75% com 144 instancias
* A taxa de erro foi de 2,25% com 4 intancias, sendo que das 4 , três vinhos pertencem a classe 2 , mas foram classificados erroneamente como 1 vinho sendo classe 1 e 2 vinhos sendo da classe 3 e o ultimo vinho restante da classe 3 foi classificado erroneamente como sendo classe 2.

1. Gerar um relatório detalhando todas as etapas desenvolvidas.

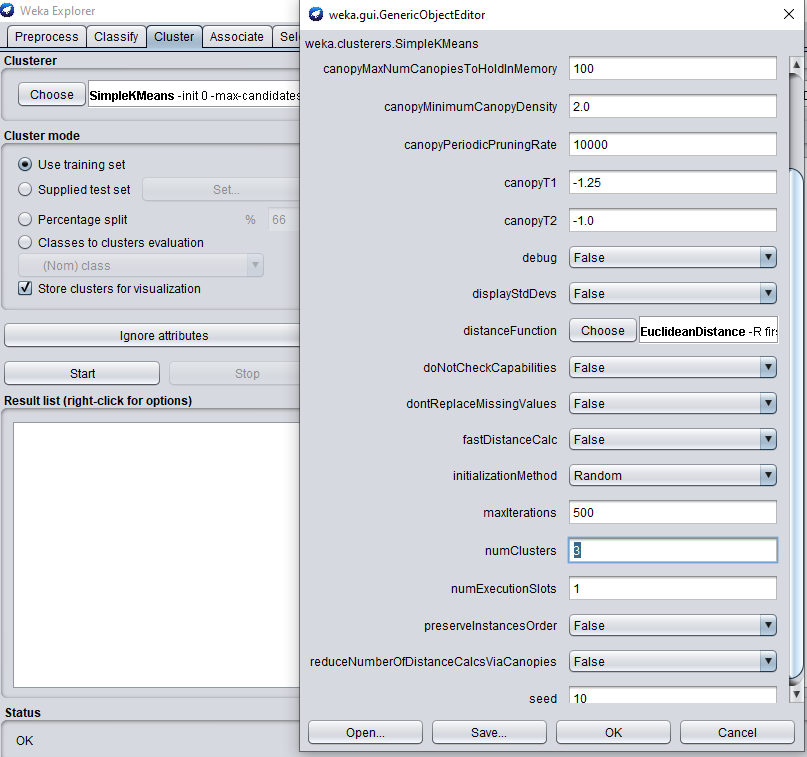
* Instalação e conhecimento sobre os objetivos e funcionaidades do software Weka
* Analise e descrição dos bancos utilizados:
* wine.arff e
* iris.arff
* Analise dos dois modelos empregados nos testes:
* De classificação: MultilayerPerceptron
* De Agrupamento: SimpleKmeans
* Analise e descrição de cada um dos modelos utilizados:
* Funções e
* Parametros;
* Execução de cada um dos modelos e analise dos resultados:
* **Execução** **K-médias para ambos os bancos:**

Passos executados para o Kmédias:

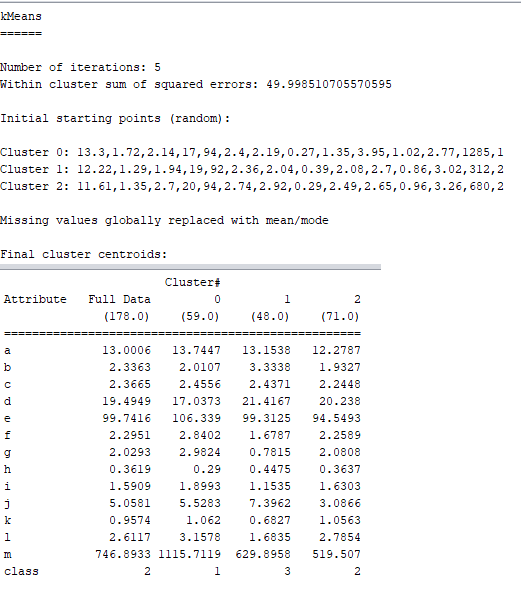
1. Definir e analizar o numero de grupos do modelo k-médias:

* Inicialmente o Numcluster = 3 e depois utilizou os valores {2, 3, 4, e 5}

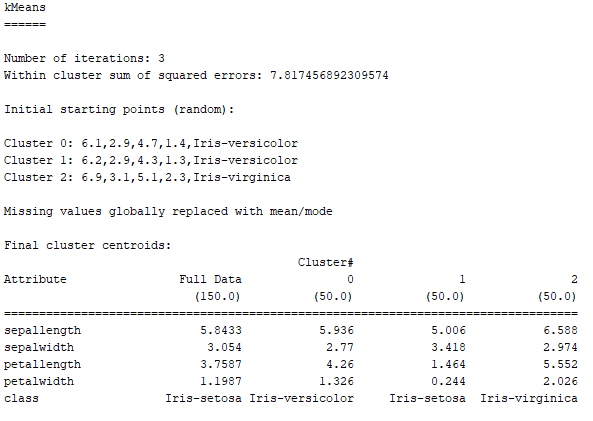
A figura a seguir mostra a definição de Numcluster = 3 (depois ok e Start):



A proxima figura mostra o agrupamento com Numcluster = 3 para base wine.arff



* A proxima figura mostra o agrupamento com Numcluster = 3 para iris.arff.



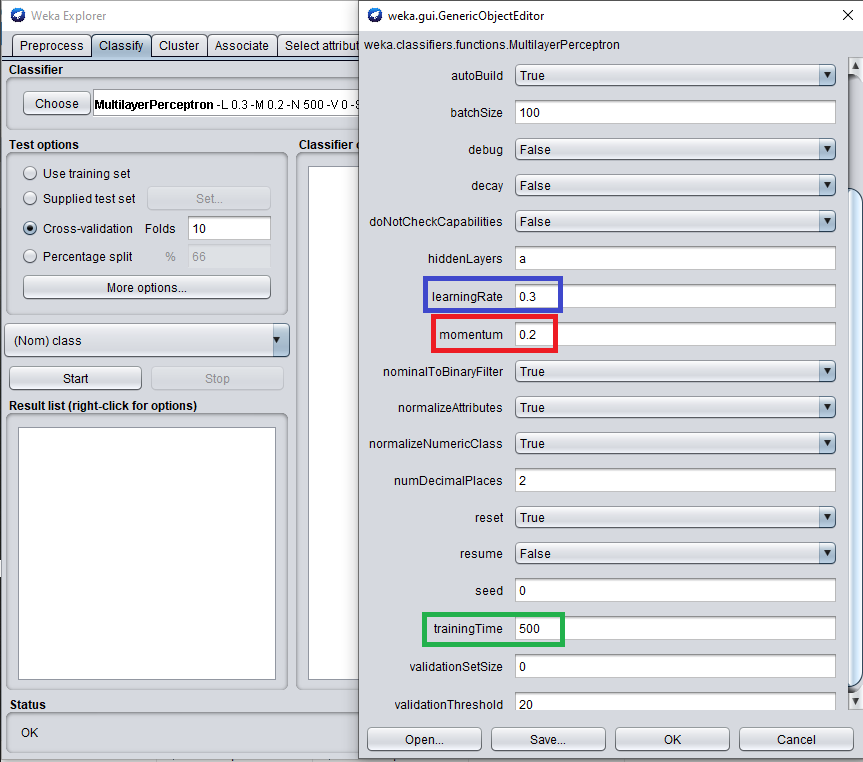
* Ao aumentar o número de grupos (fazendo ficar maior que o numero de classes) averigou-se que a soma dos erros quadráticos do cluster diminuiu, porém algumas classes ficaram subdivididas em mais grupos.

1. Variação dos centroides iniciais via parametro do **seed**:

* Para o campo seed foi fornedido os seguintes valores {2,5,10,20}
* Averigou-se que os pontos inciais escolhidos num primeiro momento podem pentecenter a um mesmo grupo e no decorrer da execução e interações os mesmos serão reagrupados, uma vez que a distância dos pontos (Euclidean distanc) ate os novos centroides definidos no decorrer das interações (pela media dos pontos do grupo) pode ser menor do que a distância eucrediana dada pelos pontos ate os centroides anteriores (iniciais), assim ocorre o reagrupamento dos pontos, inclusive dos centroides iniciais.
* **Execução** **MultilayerPerceptron para ambos os bancos:**
* Deixar selecionado a opção de teste “Cross-validation”, com 10 pastas conforme padrão do WEKA.

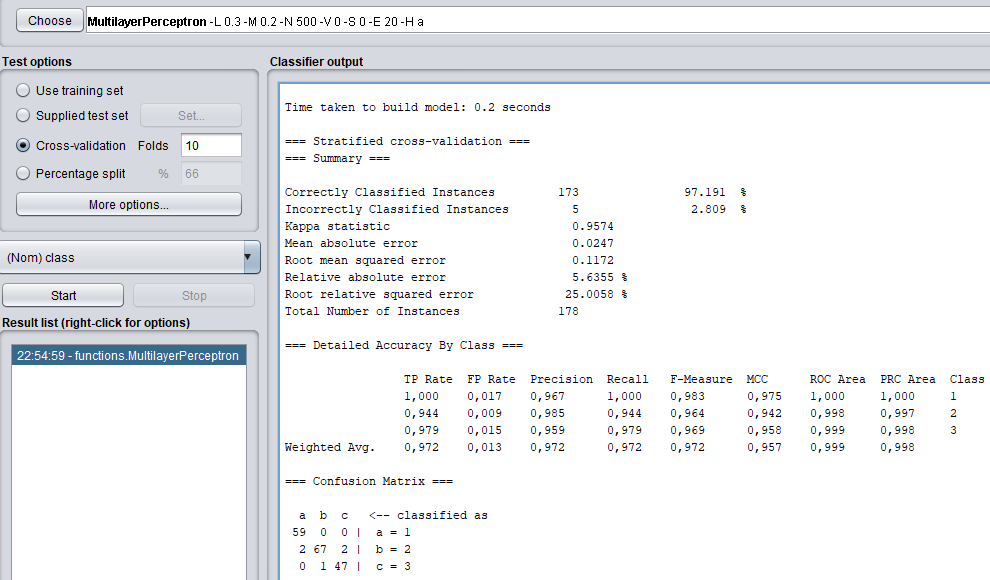
1. Modificar a taxa de aprendizado com os valores: {0,3; 0,6; 0,9} e observar os resultados.
2. Modificar o momento com os valores: {0,2 0,5; 0,8} e observar os resultados.
3. Modificar o número de épocas: {500, 1000, 2000} e observar os resultados.
4. Combinar a taxa de aprendizado e momento diferentes: utilizou-se a Taxa de Aprendizado: 0,5 e Momento: 0,1 depois com Taxa de Aprendizado: 0,1 e Momento: 0,5 (ambos experimentos com Epoca = 2000) e observar os resultados.

A proxima figura mostra os campos taxa de aprendizado (leaminingRate), momento (momentum) e número de épocas (trainingTime) a ser modificado em cada uma das bases (depois ok e Start):



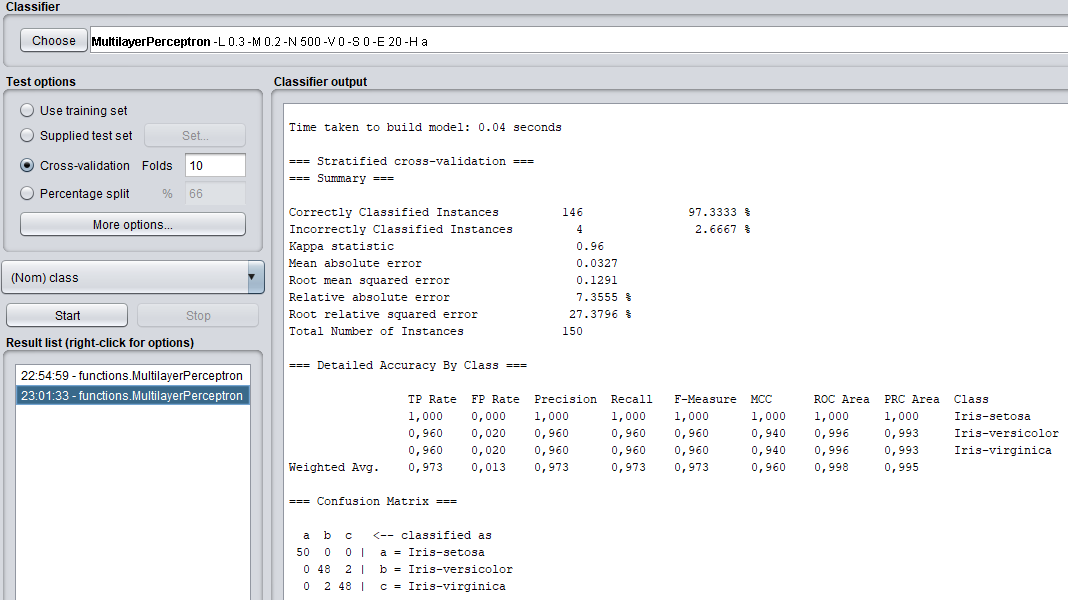
A proxima figura mostra os campos taxa de aprendizado (leaminingRate) = 0,3, momento (momentum) = 0,2 e número de épocas (trainingTime) = 500 para base **wine.arff:**

:



* Como é possivel obvervar foi obtido o seguinte resultado:
* A taxa de acerto foi de 97,191% em 173 instancias;
* A taxa de erro foi de 2,809 em 5 instancias, sendo que dois vinhos da classe 2 foram classificados como classe 1 e dois como classe 3 e ainda um vinho da classe 3 foi classificado como da classe 2.

A proxima figura mostra os campos taxa de aprendizado (leaminingRate) = 0,3, momento (momentum) = 0,2 e número de épocas (trainingTime) = 500 para base **iris.arff:**



* Como é possivel obvervar foi obtido o seguinte resultado:
* A taxa de acerto foi de 97,33%, com 146 intancias.
* A taxa de erro foi de 2,66%, com 4 instancias, sendo que duas Versicolors foram classificadas como Virgínica e duas Virgínicas foram classificadas como Versicolor.
* Apos realizar todas as alterações na taxa de aprendizado (leaminingRate), no momento (momentum) e no número de épocas (trainingTime) foi verificado que a **base ires** houve pequenas alterações, enquanto para a **base Wine** não teve mudança significativa.

1. Por fim foram realizados testes modificando o número de pastas da validação cruzada entre 2 e 5 e mantendo os valores dos outros parâmetros da rede neural no padrão do WEKA.

