Laboratório

Como usar algoritmos de aprendizado de máquina de regressão em Weka

Faça o download do dataset **housing.arff*** , e execute as seguintes tarefas:
*disponível em: www.inf.ufpr.br/menotti/am-191/data.zip

Weka tem um grande número de algoritmos de regressão disponíveis na plataforma. O grande número de algoritmos de aprendizado de máquina suportados pelo Weka é um dos maiores benefícios do uso da plataforma.
Neste post, você descobrirá como usar os melhores algoritmos de aprendizado de máquina de regressão em Weka.
Depois de ler este post você saberá:

- Aproximadamente 5 principais algoritmos de regressão suportados pelo Weka.
- Como usar algoritmos de aprendizado de máquina de regressão para modelagem preditiva em Weka.
- Sobre as principais opções de configuração dos algoritmos de regressão em Weka.

Visão Geral dos Algoritmos de Regressão

Vamos fazer um tour pelos 5 principais algoritmos de regressão em Weka. Cada algoritmo que abordamos será brevemente descrito em termos de como ele funciona, os principais parâmetros do algoritmo serão destacados e o algoritmo será demonstrado na interface do Weka Explorer.

Os 5 algoritmos que analisaremos são:

1. Regressão linear
2. K-vizinhos mais próximos
3. Árvore de Decisão
4. *Support Vector Machines*
5. *Multiple-layer Perceptron*

Estes são 5 algoritmos que você pode experimentar no seu problema de regressão como ponto de partida. Um problema padrão de regressão de aprendizado de máquina será usado para demonstrar cada algoritmo. Especificamente, o conjunto de dados de preço de casa de Boston (**housing.arff**). Cada instância descreve as propriedades de um subúrbio de Boston e a tarefa é prever os preços das casas em milhares de dólares. Existem 13 variáveis numéricas de entrada com escalas variadas descrevendo as propriedades dos subúrbios. Você pode aprender mais sobre esse conjunto de dados no UCI Machine Learning Repository .

Inicie o Weka Explorer:

1. Abra o Weka GUI Chooser.
2. Clique no botão "**Explorer**" para abrir o Weka Explorer.
3. Carregue o conjunto de dados de preços de casas de Boston no arquivo ***housing.arff***.
4. Clique em "**Classify**" para abrir a guia Classificar.

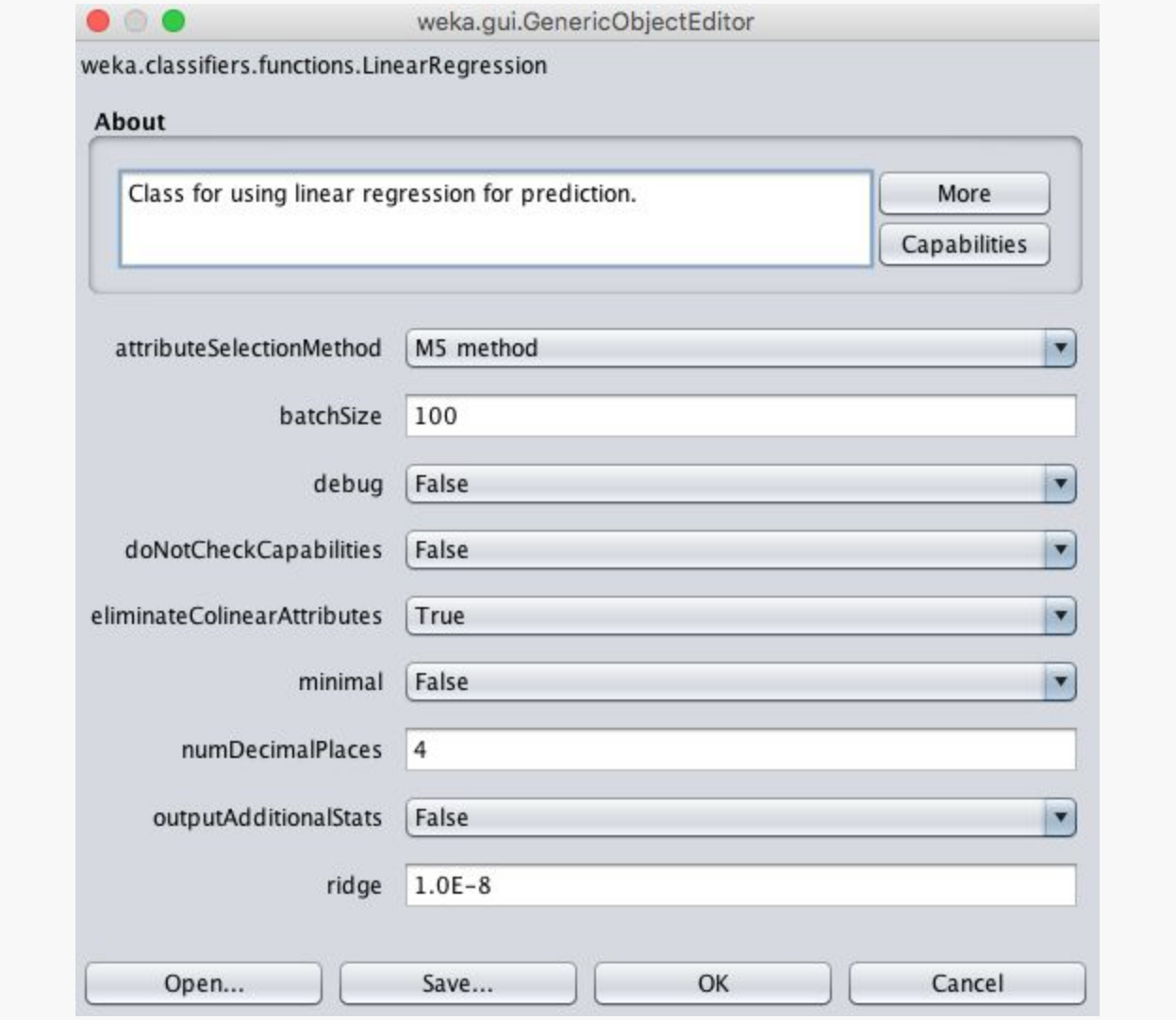
Vamos começar as tarefas olhando o algoritmo de regressão linear.

Regressão linear

A regressão linear suporta apenas problemas de tipo de regressão. Ele funciona estimando coeficientes para uma linha ou hiperplano que melhor se adapta aos dados de treinamento. É um algoritmo de regressão muito simples, rápido de treinar e pode ter um ótimo desempenho se a variável de saída para seus dados for uma combinação linear de suas entradas. É uma boa ideia avaliar a regressão linear em seu problema antes de passar para algoritmos mais complexos, caso ele tenha um bom desempenho.

Escolha o algoritmo de regressão linear:

1. Clique no botão **Choose** e selecione **Linear Regression** no grupo **functions**.
2. Clique no nome do algoritmo para revisar a configuração do algoritmo.



Weka Configuração de *Linear Regression* (Regressão Linear)

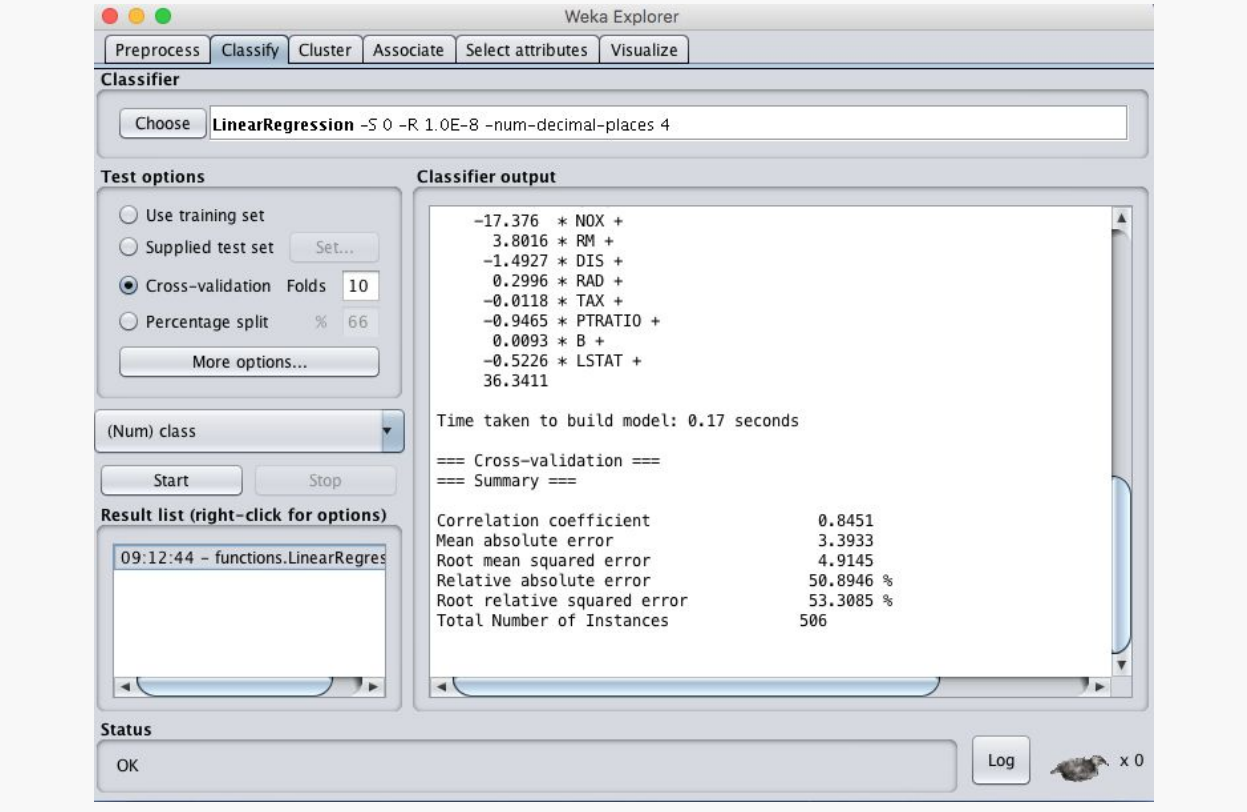
O desempenho da regressão linear pode ser reduzido se seus dados de treinamento tiverem atributos de entrada altamente correlacionados. Weka pode detectar e remover automaticamente atributos de entrada altamente correlacionados, definindo **eliminateColinearAttributes** como **True**, que é o padrão.

Além disso, os atributos não relacionados à variável de saída também podem afetar negativamente o desempenho. Weka pode executar automaticamente a seleção de recursos para selecionar apenas os atributos relevantes, definindo o **attributeSelectionMethod**. Isso é ativado por padrão e pode ser desativado.

Finalmente, a implementação do Weka usa uma técnica de regularização de cumeeira para reduzir a complexidade do modelo aprendido. Isso é feito minimizando o quadrado da soma absoluta dos coeficientes aprendidos, o que impedirá que qualquer coeficiente específico se torne muito grande (um sinal de complexidade nos modelos de regressão).

1. Clique em **OK** para fechar a configuração do algoritmo.
2. Clique no botão **Start** para executar o algoritmo no conjunto de dados de preços de casas de Boston.

Você pode ver que, com a configuração padrão, a regressão linear atinge um **RMSE** (*root mean square erro*r – raiz quadrada do erro médio quadrático) de 4,9.



Resultados Weka para Linear Regression (Regressão Linear).

k vizinhos mais próximos

O algoritmo de k-vizinhos mais próximos suporta classificação e regressão. Também é chamado de kNN. Ele funciona armazenando todo o conjunto de dados de treinamento e consultando-o para localizar os padrões de treinamento mais similares ao fazer uma previsão.

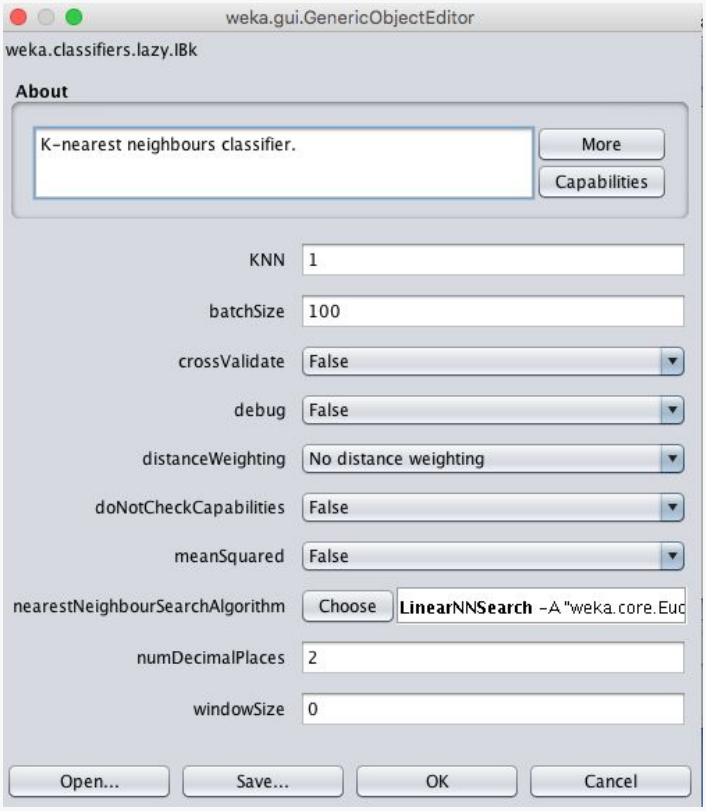
Como tal, não há outro modelo além do conjunto de dados de treinamento bruto e a única computação executada é a consulta do conjunto de dados de treinamento quando uma previsão é solicitada.

É um algoritmo simples, mas que não assume muito sobre o problema, a não ser que a distância entre as instâncias de dados é significativa ao fazer previsões. Como tal, muitas vezes consegue um desempenho muito bom.

Ao fazer previsões sobre problemas de regressão, o KNN tomará a média das instâncias mais similares no conjunto de dados de treinamento. Escolha o algoritmo KNN:

1. Clique no botão **Choose** e selecione "**IBk** no grupo **lazy**.
2. Clique no nome do algoritmo para revisar a configuração do algoritmo.

Em Weka KNN é chamado **IBk** que significa "*Instance Based k*" ou "k baseado em instância".



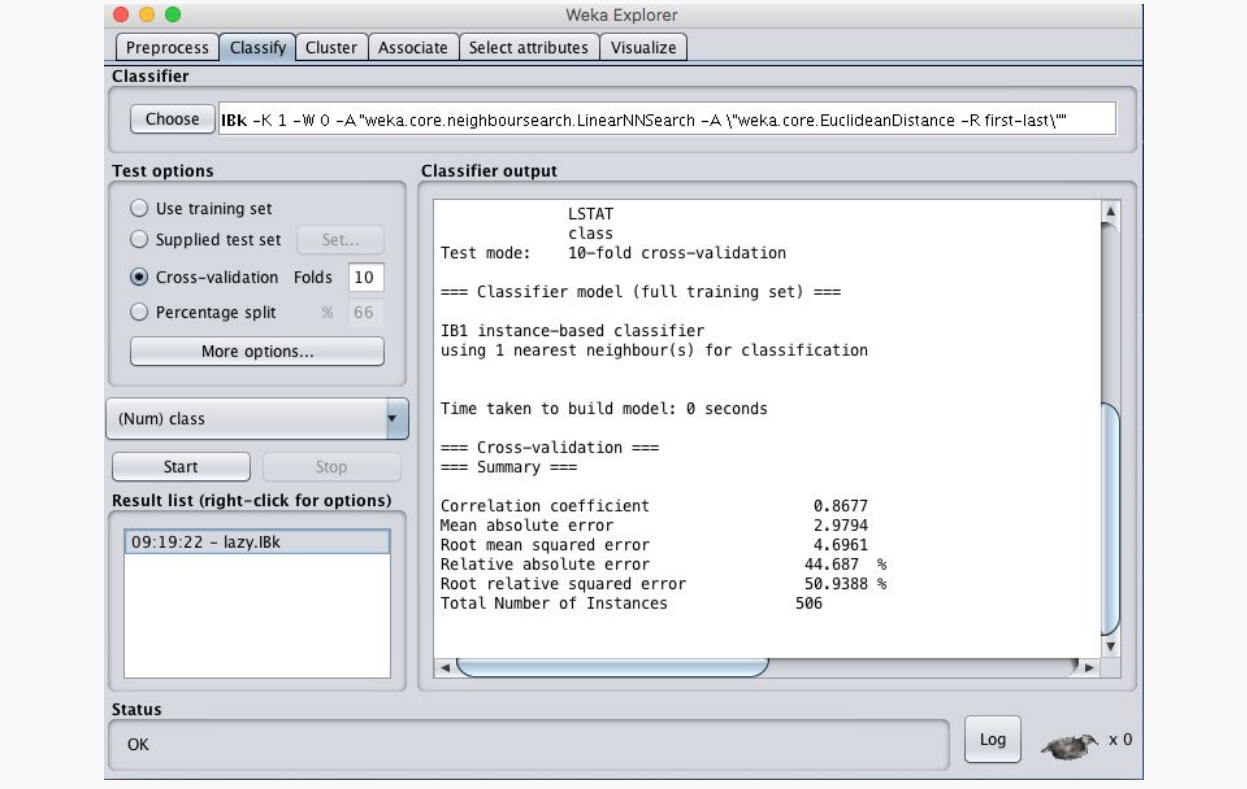
Configuração do k-Nearest-Neighbours no Weka

O tamanho da vizinhança é controlado pelo parâmetro KNN. Por exemplo, se definido como 1, as previsões serão feitas usando a instância de treinamento mais semelhante a um novo padrão para o qual uma previsão é solicitada. Valores comuns para **k** são 3, 7, 11 e 21, maiores para tamanhos de conjuntos de dados maiores. Weka pode descobrir automaticamente um bom valor para **k** usando validação cruzada dentro do algoritmo, definindo o parâmetro **crossValidate** como **True**.

Outro parâmetro importante é a medida de distância usada. Isso é configurado no **nearestNeighbourSearchAlgorithm** mais próximo, que controla o modo como os dados de treinamento são armazenados e pesquisados. O padrão é um **LinearNNSearch**. Clicar no nome desse algoritmo de pesquisa fornecerá outra janela de configuração na qual você poderá escolher um parâmetro **distanceFunction**. Por padrão, a distância euclidiana é usada para calcular a distância entre instâncias, o que é bom para dados numéricos com a mesma escala. Distância de Manhattan (**Manhattan Distance**) é boa para usar se seus atributos diferem em medidas ou tipo.
É uma boa ideia tentar um conjunto de diferentes valores de k e medidas de distância em seu problema e ver o que funciona melhor.

1. Clique em **OK** para fechar a configuração do algoritmo.
2. Clique no botão **Start** para executar o algoritmo no conjunto de dados de preços de casas de Boston.

Você pode ver que, com a configuração padrão, o algoritmo **KNN** atinge um **RMSE** de 4,6.



Resultados da Regressão Weka para o Algoritmo k-Nearest Neighbours

Árvore de Decisão

As árvores de decisão podem suportar problemas de classificação e regressão. Árvores de decisão são mais recentemente referidas como Árvores de Classificação e Regressão ou (**Classification and regression tree** - CART). Elas trabalham criando uma árvore para avaliar uma instância de dados, começar na raiz da árvore e mover a cidade para as folhas (raízes porque a árvore é desenhada com uma perspectiva invertida) até que uma previsão possa ser feita. O processo de criação de uma árvore de decisão funciona selecionando avidamente o melhor ponto de divisão para fazer previsões e repetir o processo até que a árvore tenha uma profundidade fixa.

Depois que a árvore é construída, ela é removida para melhorar a capacidade do modelo de generalizar para novos dados.
Escolha o algoritmo da árvore de decisão:

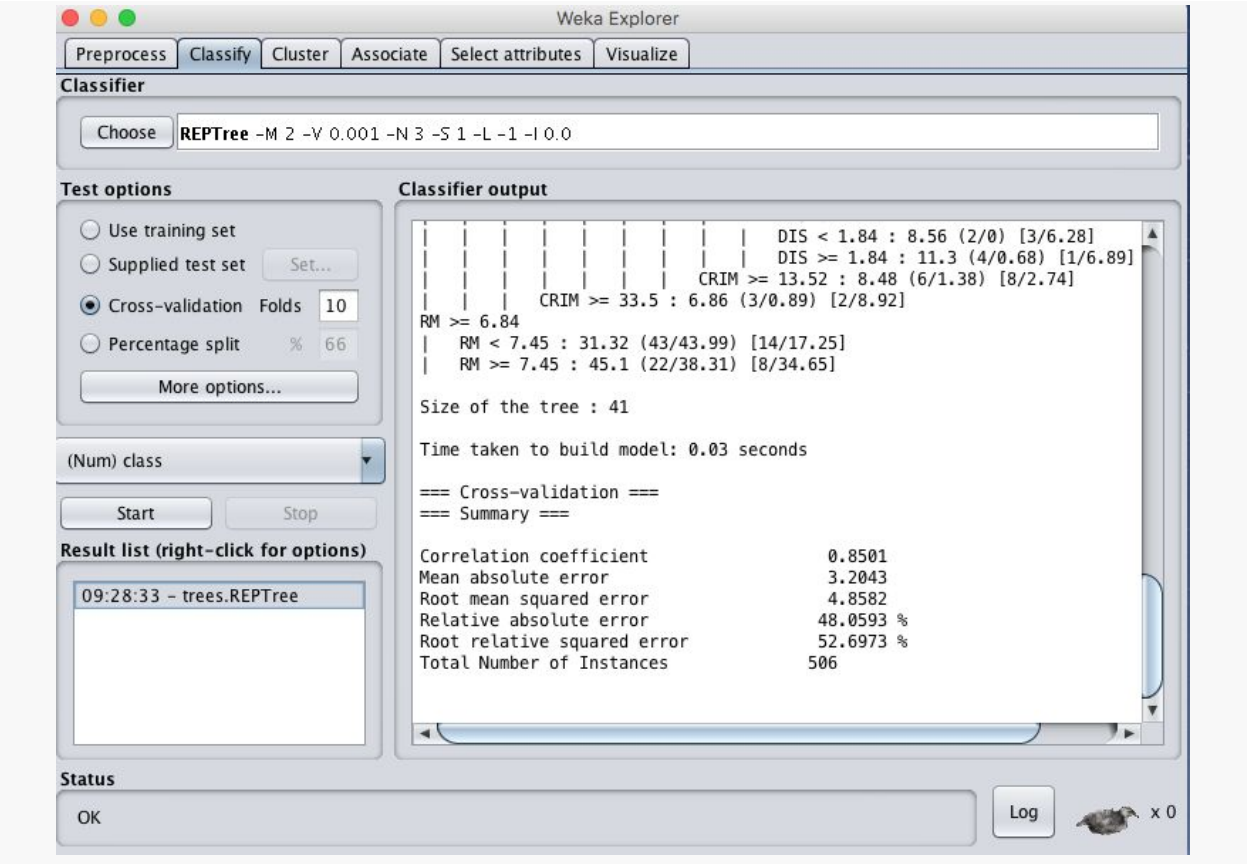
1. Clique no botão **Choose** e selecione **REPTree** no grupo **trees**.
2. Clique no nome do algoritmo para revisar a configuração do algoritmo.

Configuração Weka para Algoritmo de Árvore de Decisão

A profundidade da árvore é definida automaticamente, mas pode especificar uma profundidade no atributo **maxDepth**. Você também pode optar por desativar a remoção definindo o parâmetro **noPruning** como **True**, embora isso possa resultar em pior desempenho. O parâmetro **minNum** define o número mínimo de instâncias suportadas pela árvore em um nó folha ao construir a árvore a partir dos dados de treinamento.

1. Clique em **OK** para fechar a configuração do algoritmo.
2. Clique no botão **Start** para executar o algoritmo no conjunto de dados de preços de casas de Boston.

Você pode ver que, com a configuração padrão, o algoritmo da árvore de decisão atinge um **RMSE** de 4,8.



Resultados da Regressão Weka para o Algoritmo da Árvore de Decisão

Regressão por *Support Vector Machines* (Máquinas de Vetores de Suporte)

As *Support Vector Machines* foram desenvolvidas para problemas de classificação binária, embora tenham sido feitas extensões à técnica para suportar problemas de classificação e regressão de várias classes. A adaptação do SVM para regressão é chamada de Regressão Vetorial de Suporte ou SVR (*Support Vector Regression*) para abreviar.

O **SVM** foi desenvolvido para variáveis de entrada numérica, embora converta automaticamente valores nominais em valores numéricos. Os dados de entrada também são normalizados antes de serem usados.

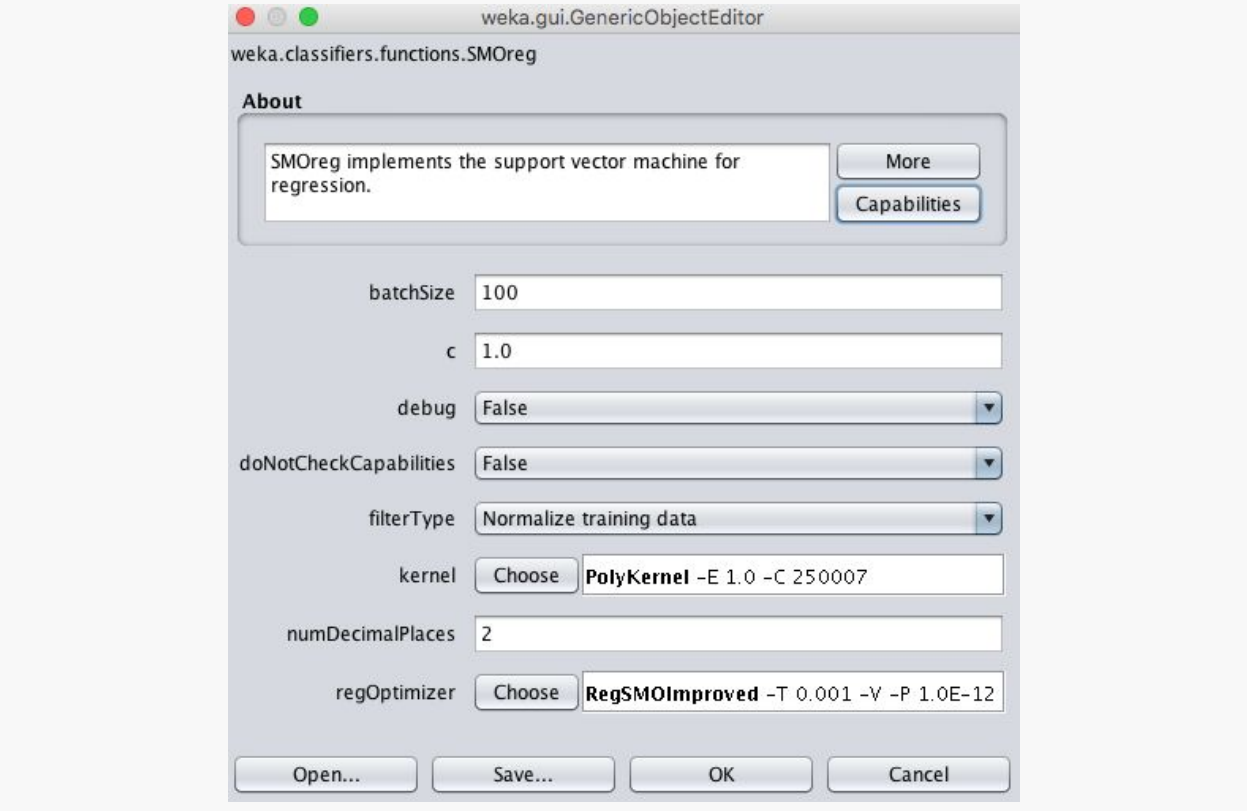
Ao contrário do **SVM**, que encontra uma linha que separa melhor os dados de treinamento em classes, o **SVR** trabalha encontrando uma linha de melhor ajuste que minimiza o erro de uma função de custo. Isso é feito usando um processo de otimização que considera apenas as instâncias de dados no conjunto de dados de treinamento que estão mais próximas da linha com o custo mínimo. Essas instâncias são chamadas de **vetores de suporte**, daí o nome da técnica.

Em quase todos os problemas de interesse, uma linha não pode ser desenhada para melhor encaixar os dados, portanto, uma margem é adicionada ao redor da linha para relaxar a restrição, permitindo que algumas previsões ruins sejam toleradas, mas permitindo um melhor resultado geral.

Finalmente, poucos conjuntos de dados podem ser ajustados apenas com uma linha reta. Às vezes, uma linha com curvas ou mesmo regiões poligonais precisa ser marcada. Isso é conseguido projetando os dados em um espaço dimensional superior para desenhar as linhas e fazer previsões. Kernels diferentes podem ser usados para controlar a projeção e a quantidade de flexibilidade.

Escolha o algoritmo SVR:

1. Clique no botão **Choose** e selecione **SMOreg** no grupo **function**.
2. Clique no nome do algoritmo para revisar a configuração do algoritmo.



Configuração Weka para o Algoritmo de Regressão Vetorial de Suporte

O parâmetro **C**, chamado de parâmetro de complexidade (ou regularização) no Weka , controla quão flexível pode ser o processo para desenhar a linha para ajustar os dados. Um valor de 0 não permite violações da margem, enquanto o padrão é 1.

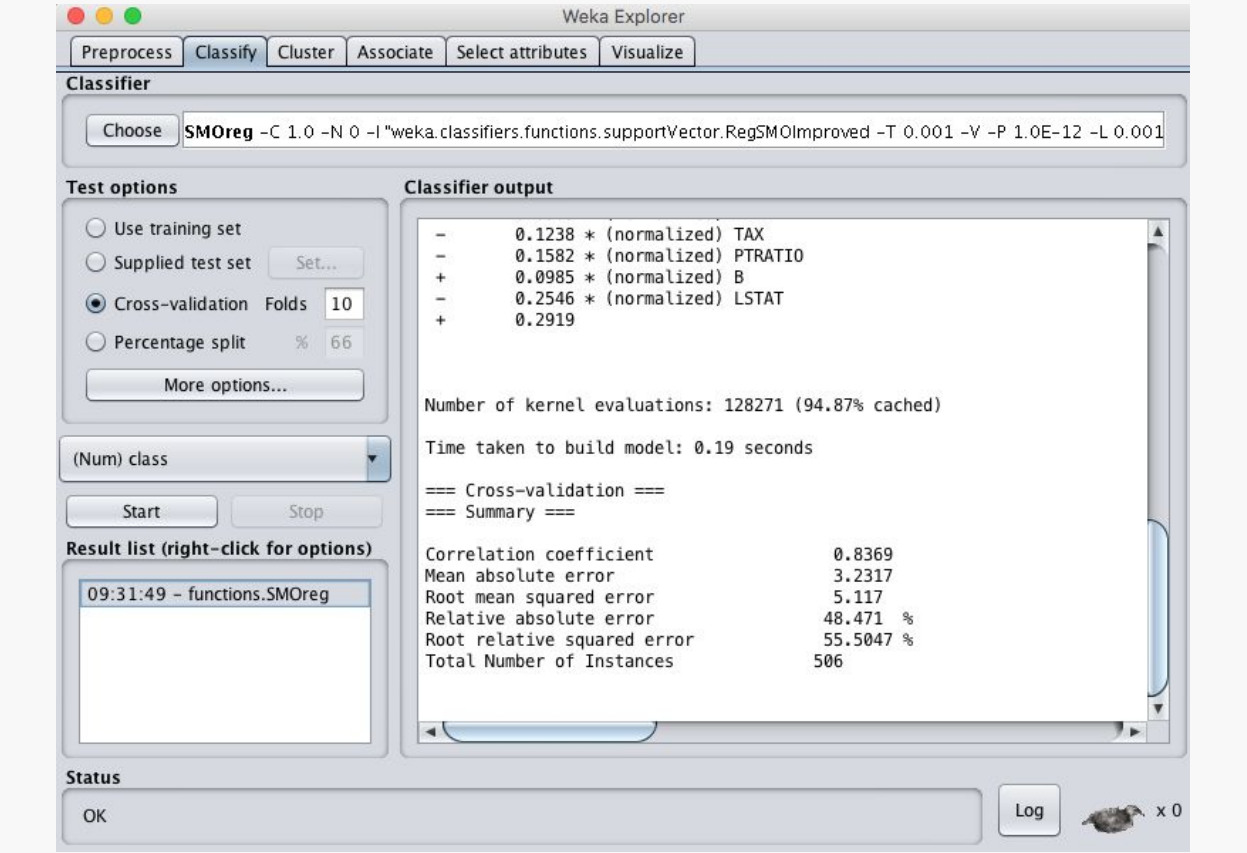
Um parâmetro chave no **SVM** é o tipo de kernel a ser usado. O kernel mais simples é um kernel linear que separa os dados com uma linha reta ou hiperplano. O padrão em Weka é um Kernel Polinomial que irá ajustar os dados usando uma linha curva ou "contorcida" (*wiggly*), quanto maior o polinômio, mais contorcida é a curva (o valor do expoente).

O Kernel Polinomial tem um expoente padrão de 1, o que o torna equivalente a um kernel linear. Um kernel popular e poderoso é o Kernel **RBF** (*Radial Basis Function*) ou o Kernel de Função de Base Radial que é capaz de aprender polígonos fechados e formas complexas para se ajustar aos dados de treinamento.

É uma boa idéia tentar um conjunto de kernels diferentes e valores C (complexidade / regularização) em seu problema e ver o que funciona melhor.

1. Clique em **OK** para fechar a configuração do algoritmo.
2. Clique no botão **Start** para executar o algoritmo no conjunto de dados de preços de casas de Boston.

Você pode ver que, com a configuração padrão, o algoritmo SVR atinge um RMSE de 5,1.



Resultados de regressão Weka para o algoritmo de regressão vetorial de suporte

Perceptron Multicamadas (Multi-layer Perceptron)

Os algoritmos de **Perceptron Multi-Layer** (*Multi-layer Perceptron*) suportam tanto os problemas de regressão quanto de classificação. É também chamado de redes neurais artificiais ou simplesmente redes neurais para breve. As redes neurais são um algoritmo complexo para usar na modelagem preditiva, pois há muitos parâmetros de configuração que só podem ser ajustados de forma eficaz por meio da intuição e de muitas tentativas e erros.

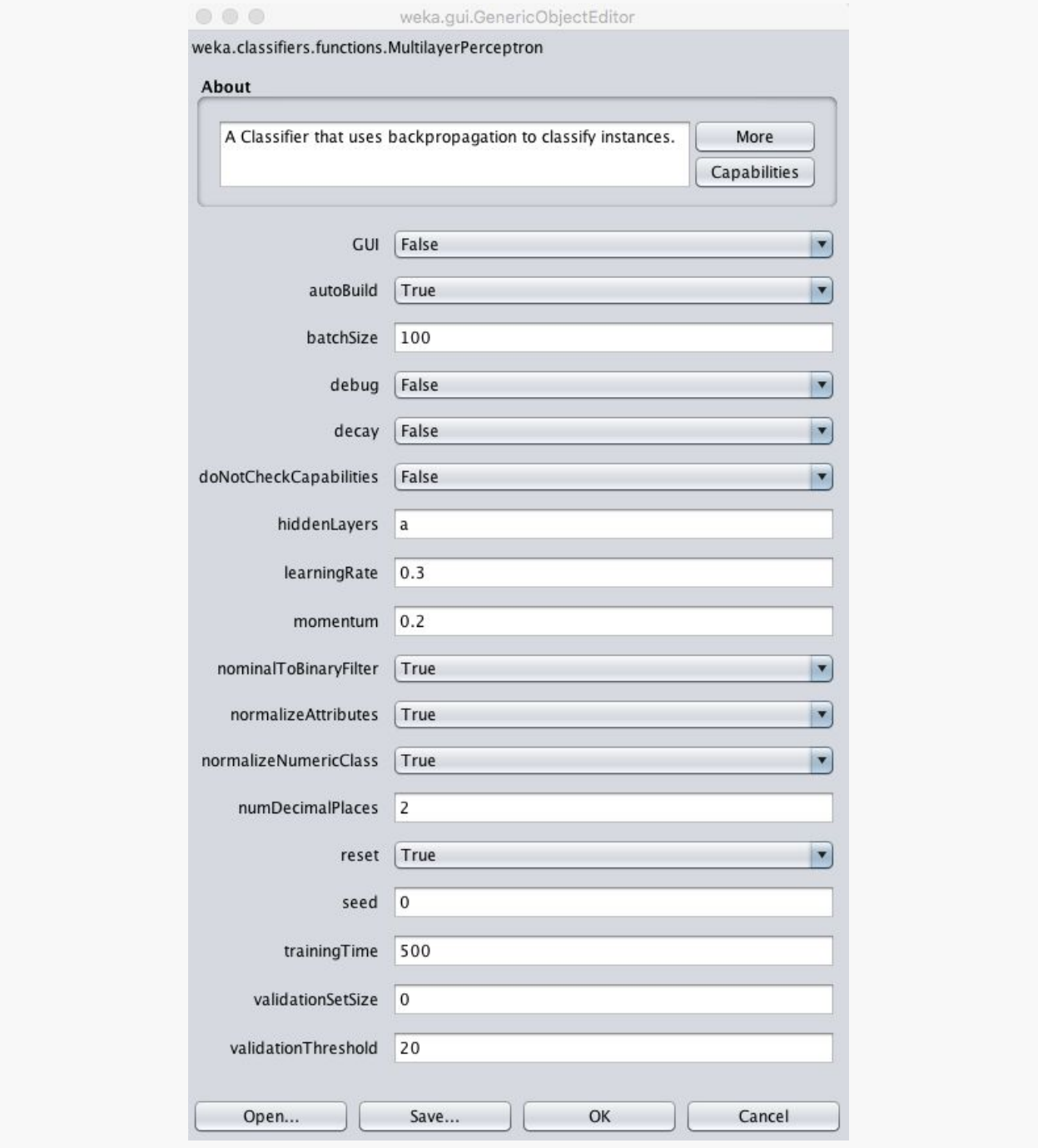
É um algoritmo inspirado em um modelo de redes neurais biológicas no cérebro onde pequenas unidades de processamento chamadas neurônios são organizadas em camadas que, se bem configuradas, são capazes de se aproximar de qualquer função. Na classificação, estamos interessados em aproximar a função subjacente para melhor discriminar entre classes. Em problemas de regressão, estamos interessados em aproximar uma função que melhor se ajuste à saída do valor real.

Escolha o algoritmo Perceptron Multi-Layer:

1. Clique no botão **Choose** e selecione **MultilayerPerceptron** sob o grupo **function**.
2. Clique no nome do algoritmo para revisar a configuração do algoritmo.

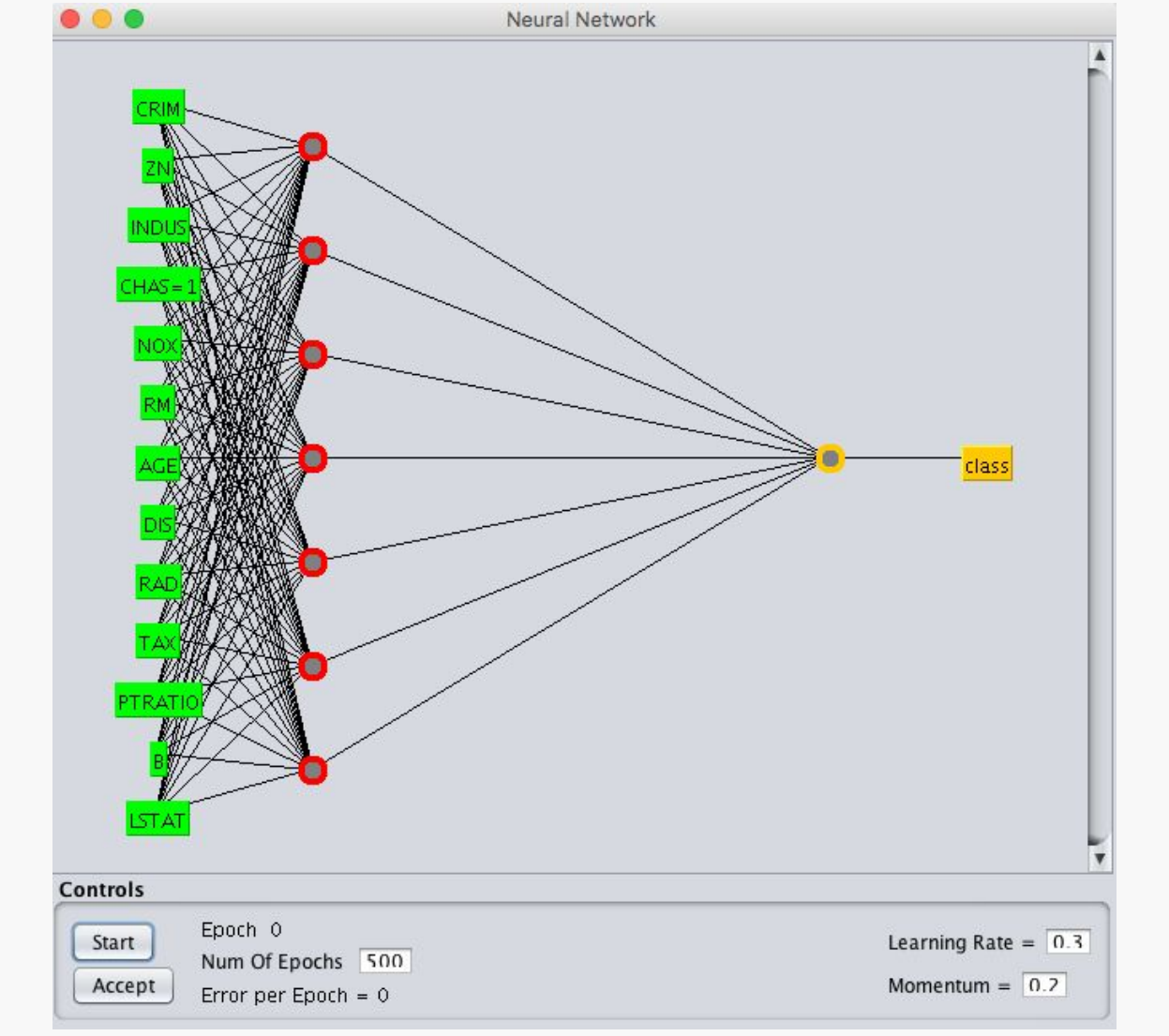
Você pode especificar manualmente a estrutura da rede neural usada pelo modelo, mas isso não é recomendado para iniciantes.

O padrão irá projetar automaticamente a rede e treiná-la no seu conjunto de dados. O padrão criará uma rede com uma única camada oculta. Você pode especificar o número de camadas ocultas no parâmetro **hiddenLayers**, definido como "**a**" automático por padrão.



Configuração Weka para o Algoritmo Perceptron Multi-Layer

Você também pode usar uma GUI para projetar a estrutura de rede. Isso pode ser divertido, mas é recomendável usar a GUI com um simples train e teste de divisão de seus dados de treinamento, caso contrário, você será solicitado a projetar uma rede para cada uma das 10 execuções de validação cruzada.



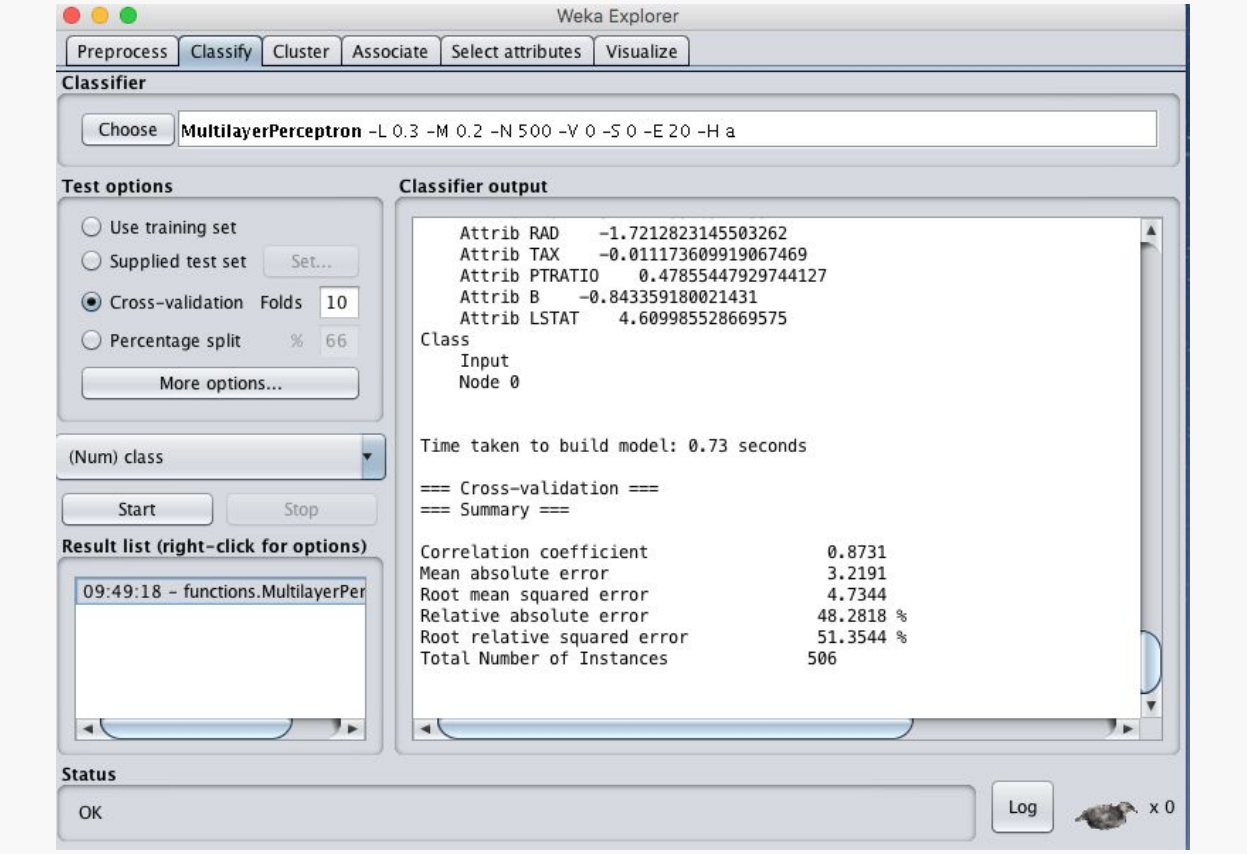
Weka GUI Designer para o Algoritmo Perceptron Multi-Layer

Você pode configurar o processo de aprendizado especificando o quanto atualizar o modelo em cada época, definindo a **taxa de aprendizado** (*learning rate*). valores comuns são pequenos, como valores entre 0,3 (o padrão) e 0,1.

O processo de aprendizado pode ser ajustado com um **momentum** (definido como 0,2 por padrão) para continuar atualizando os pesos mesmo quando nenhuma alteração precisar ser feita, e um decaimento (defina o decaimento como True), o que reduzirá a taxa de aprendizado ao longo do tempo. mais aprendizagem no início do treinamento e menos no final.

1. Clique em **OK** para fechar a configuração do algoritmo.
2. Clique no botão **Start** para executar o algoritmo no conjunto de dados de preços de casas de Boston.

Você pode ver que, com a configuração padrão, o algoritmo **Multi-Layer Perceptron** atinge um **RMSE** de 4,7.



Resultados do Algoritmo de Regressão de Multiple-layer Perceptron no Weka

Resumo

Neste tutorial você descobriu algoritmos de regressão em Weka.

Especificamente você aprendeu:

- Cerca de 5 algoritmos de regressão que você pode usar para modelagem preditiva.
- Como executar algoritmos de regressão em Weka.
- Sobre as principais opções de configuração para algoritmos de regressão em Weka.

Referências

Jason Brownlee em 22 de julho de 2016 em **Weka Machine Learning**
How To Use Regression Machine Learning Algorithms in Weka
https://machinelearningmastery.com/use-regression-machine-learning-algorithms-weka/

Jason Brownlee em 10 de Agosto de 2016 em **Weka Machine Learning**
How to Work Through a Regression Machine Learning Project in Weka Step-By-Step
https://machinelearningmastery.com/regression-machine-learning-tutorial-weka/

Ian H. Witten em 29 de Setembro de 2031 em **Weka**
Linear Regression - Data Mining with Weka
https://www.youtube.com/watch?v=6tDnNyNZDF0