**Árvore de decisão no WEKA**

Abra o Weka e clique em **Explorer**

Vamos carregar a base de dados **weather.nominal.arff**. Essa base vem junto com o Weka, para testes.

Essa base contém os atributos temperatura, vento,umidade e aspecto.

A classe se vai ter jogo ou não (sim ou não) como demos o exemplo acima, porém, os dados aqui estão em inglês

Para abrir clique em **Open file**… navegue até a pasta de instalação do weka, em seguida abra a pasta **data,**clique no arquivo informado acima. Como no gif abaixo:

Após carregar o dataset, é possível ver informações da base. Podemos ver que esta contém 4 atributos (***outlook, temperature,humidy,windy***) e por fim a coluna ‘Play’ que pode ser da classe Yes ou No.

Para rodar o algoritmo clique na aba **Classify**, em seguida clique em **Choose**para escolher o algoritmo.

Será exibida diversas categorias, clique em **trees** e escolha o algoritmo **J48**, como no gif abaixo:

Ok, com o algoritmo selecionado podemos iniciar a classificação clicando em Start.

Mas antes disso, vamos entender melhor essa tela.

Cada algoritmo contém **parâmetros** que o usuário pode definir antes da execução do mesmo.

Clique em cima do algoritmo para visualizar as opções de parâmetros disponíveis neste.

Perceba que para esse algoritmo existem diversos parâmetros, veja:

Por enquanto mantenha os valores default.

Na aba test options, temos as formas de teste e validação que iremos aplicar.

Por padrão o Weka, sugere aplicar o validação usando a técnica de **Cross-Validation** com 10 folds.

Para entender melhor como essa técnica funciona, leia [esse](http://minerandodados.com.br/index.php/2017/03/15/analise-de-sentimentos-twitter-como-fazer/#avaliando-modelo) artigo que fizemos onde falamos mais dessa técnica

Clique em **Start** para executar o algoritmo.

Ao clicar em Start o algoritmo é executado e validado usando a opção de teste definida.

È exibido os resultados do processo de classificação.

Nesse caso **50%** das instâncias foram calculadas corretamente.

Podando a Àrvore

Podemos realizar diversos testes e o Weka irá salvar os resultados na lista de resultados na parte inferior a esquerda.

Vamos executar o algoritmo novamente

Porém, agora iremos alterar o valor do parâmetro **confidence factor** para 1 e ver o que acontece.

O Weka manteve o último resultado em buffer e criou uma nova tela de resultados, quanto clicamos em Start.

Podemos ver que o percentual de instâncias classificadas corretamente **aumentou( 57% )**, após a alteração do parâmetro **confidence factor**.

Isso aconteceu porque esse parâmetro, é usado para fazer uma poda na árvore de decisão.

Essa poda significa diminuir o tamanho da árvore usando atributos que não são tão importantes para árvore

Dessa forma a árvore fica **menor** e mais **genérica**, ou seja, ela não está tão idêntica aos dados de treino.

O que pode ser bom para classificação de instâncias que esta nunca viu.

No nosso caso, melhorou a classificação após aumentar o parâmetro **confidence factor** para **1**

Com essa alteração estamos dizendo para o modelo que queremos uma árvore mais genérica.

Nessa tela é interessante também visualizar a árvore que o modelo gerou,

Para isso, clique com o botão direito do mouse em um resultado na lista e em seguida clique em **Visualize tree**

**SVM – Suporte Vector Machines**

O Svm é um algoritmo muito famoso e bastante usado.

Este algoritmo através dos dados de treino, cria uma “linha” que divide as classes conforme características encontradas nos dados.

Essa linha normalmente é chamda de **“Hyperplano”**

Imagine uma base que contém 2 classes a serem classificadas.

Após conseguir dividir os dados (classe A de um lado e classe B de outro lado)

É criada uma margem que é a distância entre essas duas classes.

O SVM possui um parâmetro C que basicamente dita como será construída essa margem que separa as classes.

Valores pequenos no parâmetro C diz para o SVM que o modelo deve ser **menos**tolerante a erros

Por consequência, o modelo irá tentar separar **melhor** os dados de treino.

Um valor maior do parâmetro C diz para o SVM que o modelo pode ser mais tolerante a erros

Isso significa que o modelo permitirá mais erros, e a margem será menor entre as classes.

Para rodar o SVM usando Weka, clique no botão Choose e escolha o classificador chamado **SMO**que fica na categoria **functions,**veja:

Após escolher o classificador vamos manter as configurações padrão e rodar o algoritmo

Após execução a tela de resultados exibe as métricas.

Esse se saiu melhor perante a árvore de decisão, **64%** de instâncias corretas.

O valor atual do parâmetro C é **1.0**,

Tuning do SVM

Vamos alterar o parâmetro C do SVM para **1.5** e ver como este se comporta.

Isso é bem fácil de fazer usando o Weka, veja:

Como podemos perceber, ao aumentar o valor do parâmetro C o modelo piorou, o número de classificações corretas diminuiu (**57%**)

Isso aconteceu, pois, aumentamos a margem, assim o modelo permitiu mais erros nos dados de treino.

Por consequência, nosso modelo ficou menos complexo e na hora do teste, este se saiu pior.

O Weka permite fazer diversos ajustes nos algoritmos, e melhor, é possível testar de forma rápida e intuitiva. Faça mais testes

**Conclusão**

Nesse artigo vimos o poder do Weka.

Com o Weka podemos fazer inúmeras tarefas de forma rápida, o que aumenta a nossa produtividade como cientista de dados.

Principalmente quando queremos experimentar, essa ferramenta é perfeita.

Vimos que com o Weka é possível consultar dados de vários SGBD’s, isso permite trabalhar com fontes de dados diferentes.

Outro recurso fantástico do Weka, são os diversos filtros para pré-processamento e transformação de dados. Tudo isso acessível com **poucos cliques!**

Vimos também que experimentar vários algoritmos de Machine learning é muito simples

Inclusive é possível parametrizar algoritmos e comparar os resultados.

Se você gostou desse artigo, não se esqueça de assinar nossa lista de e-mails e **compartilhar** com seus amigos!

Vamos criar uma comunidade de mineradores