

APRENDIZAGEM AUTOMÁTICA

**RELATÓRIO DE ESTUDO SOBRE OS EXEMPLARES DE CONJUNTOS DE DADOS ESTUDADOS NO WEKA**





APRENDIZAGEM AUTOMÁTICA

**RELATÓRIO DE ESTUDO SOBRE OS EXEMPLARES DE CONJUNTOS DE DADOS ESTUDADOS NO WEKA**

**Elaborado por:**

**Venâncio Ezequias Sapal**

**&**

**Zinga Firmino René**

**Professores:**

**Phd. Lázaro Makili**

**&**

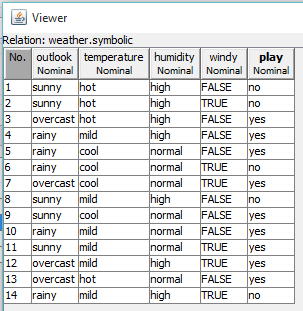
**Moisés Ferreira**

**INTRODUÇÃO**

**Resposta Da Tarefa 1:**

Através do painel *Preprocess* carregando o conjunto de dados denominado ***weather.nominal****.****arff*.** Este contém uma versão do conjunto de dados *weather* com todos os atributos nominais.

O conjunto possui 14 instâncias e 5 atributos (*outlook*, *temperature*, *humidity*, *windy* e *play*), constituindo este último o atributo classe (*yes*, *no*). Clicando no nome de um atributo no subpainel da esquerda para ver a informação acerca dos mesmos no subpainel da direita observou-se seguinte:



O atributo outlook possui 3 possíveis valores entre os 14 temos 5 sunny, 4 overcast e 5 rainy. O atributo temperature possui 3 possíveis valores entre eles 4 hot 6 mild e 6 cold. O atributo humidity possui dois possiveis valores entre os 14 temos 7 high e 7 normal. O atributo windy possui dois possíveis valores entre os 14 temos 6 true e 8 false. O atributo play possui dois possíveis valores entre os 14 temos 9 yes e 5 no.

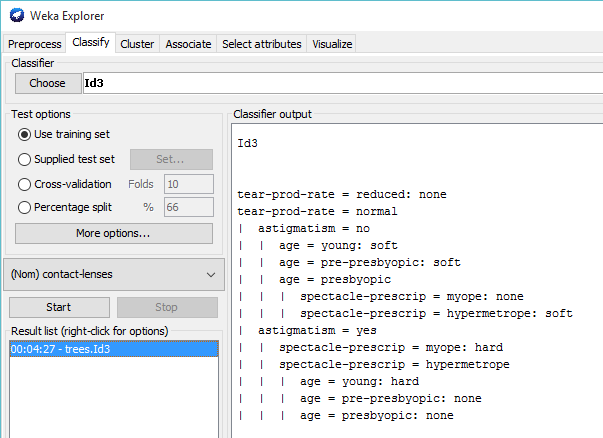
Carregando o conjunto de dados denominado ***iris.arff*.** Conjunto contém um total de possui 150 instâncias e 5 atributos (*sepallength*, *sepalwidth*, *petallength*, *petalwidth* e *class*), constituindo os atributos observou-se que os atributos são nominais, o atributo petallength possui como valor mínimo 1 e valor máximo 6,9.

Na possibilidade de ver e editar dados a partir do Weka. Para tal carregando novamente o ficheiro ***weather.nominal.arff*.** Clicando o botão *Edit* na linha de botões do painel de preprocessamento. Com isto abre-se a janela do visualizador/editor (*viewer*), mostrando todas as instâncias do conjunto de dados. A função da primeira coluna no visualizador é de ordenar as instâncias, A classe da instância 8 do conjunto de dados é *no*.

Carregando o conjunto de dados *iris.arff* e abrindo no editor. O conjunto possui 4 atributos numéricos e 1 atributo nominal.

**Resposta Da Tarefa 2:**

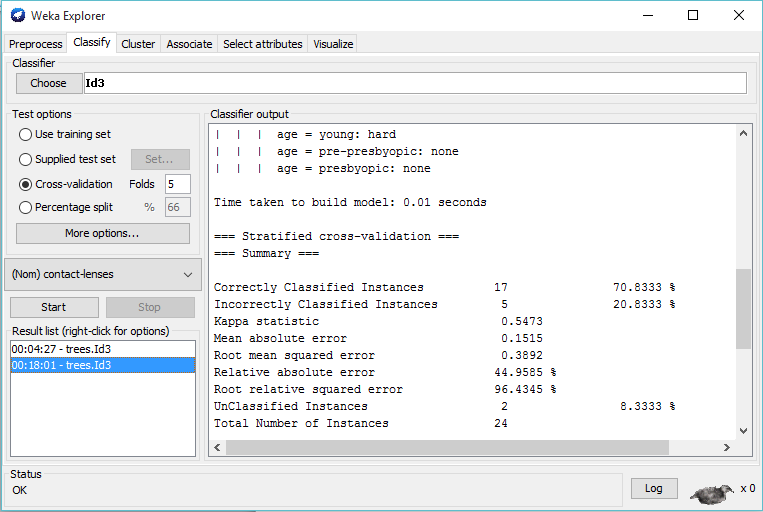
Uma das vantagens das árvores de decisão é a sua "interpretabilidade", ou seja, as regras aprendidas para classificação são fáceis de ser seguidas por uma pessoa, a diferença de muitos métodos que funcionam como "caixas pretas". Trataremos de ver isso utilizando dois conjuntos de dados, *contact-lens* e *credit*.

No explorador do Weka, abrindo o ficheiro *contact-lens.arff*. Começamos primeiro por treinar o algoritmo ID3 com o referido conjunto de dados. Para tal, no separador *classify*, seleccionamos *classifiers > trees > ID3*. Seleccionando também *Use training set* na secção *Test options* e clicando no botão *start* obtemos o seguinte resultado:

Observamos os resultados apresentados na janela da direita e tratamos de seguir a árvore de decisão aprendida. Constatamos que é compreensível a referida árvore.

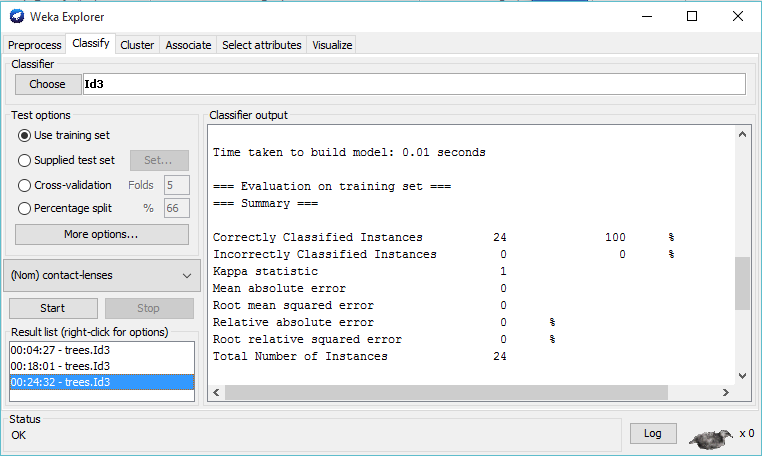
Observou-se que classificação atribuída pela árvore à instância com os valores de atributos *<age = young, spectacle-prescrip = myope, astigmatism = no, tear-prod-rate = normal> é normal.*

Treinando novamente a árvore de decisão utilizando como método de teste a validação cruzada em 5 folhas. Para tal, na secção *Test Options*, seleccionamos *crossvalidation* e *folds* = 5. Observando a quantidade (ou percentagem) de instâncias correctamente classificadas e comparando com o caso anterior, ao avaliar a taxa de erro observou-se o seguinte:



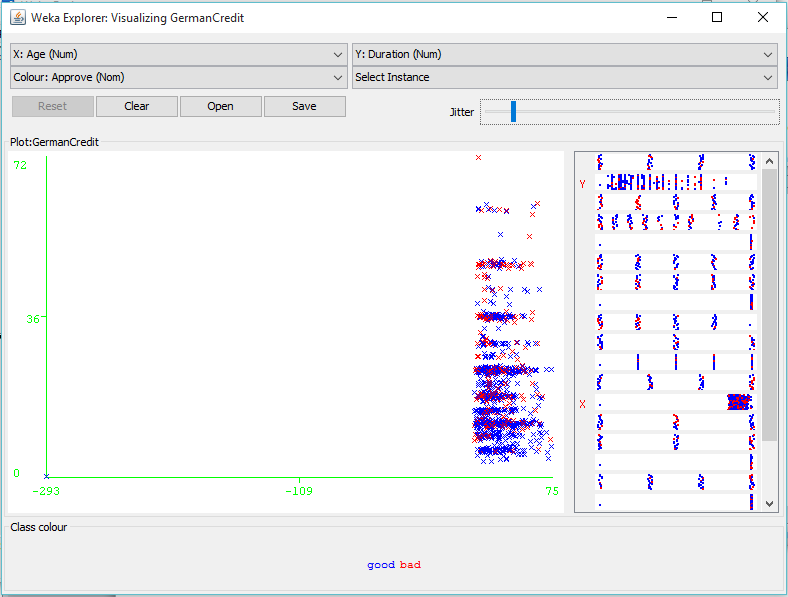
17 das instancias observadas foram correctamente classificadas o que corresponde o que corresponde a 70,8333 % das instancias e 5 das instancias observadas foram não foram correctamente classificadas o que corresponde 20,8333 % do total de intancias.

Ao passo que ao usarmos o método de teste use training set observou-se o seguinte:



24 das instancias observadas foram correctamente classificadas o que corresponde o que corresponde a 100 % das instancias e 0 das instancias observadas foram nao foram correctamente classificadas o que corresponde 0 % do total de intancias. Isso significa que para treinar o algoritmo ID3 com o referido conjunto de dados o método mais adequado é o use training set porque possui uma taxa de erro menor que todas de 0 %.

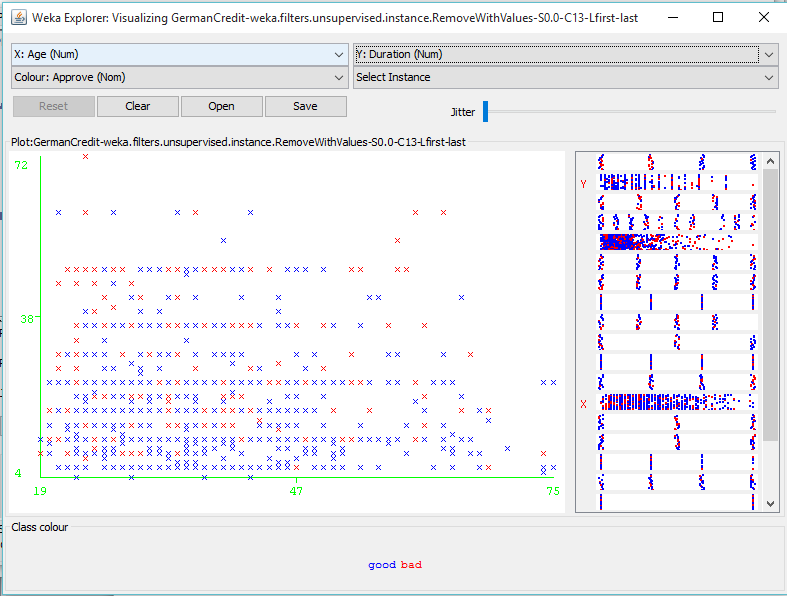
Carregamos o ficheiro *credit.arff*. conjunto que descreve, a partir de um conjunto de atributos, as pessoas como "boas" ou "de risco" para a concessão de crédito. Ao utilizar um conjunto de dados, geralmente é boa ideia visualizá-lo primeiro. indo para a secção *Visualize*. Clicamos em qualquer dos gráficos para abrir uma nova janela que mostra um gráfico para dois atributos dados. Visualizamos o gráfico para os atributos *age* e *duration*. O inusual que oservamos é que o atributo “Age” possui valores negativos, Demos um click sobre qualquer ponto para ver todos os valores dos seus atributos.

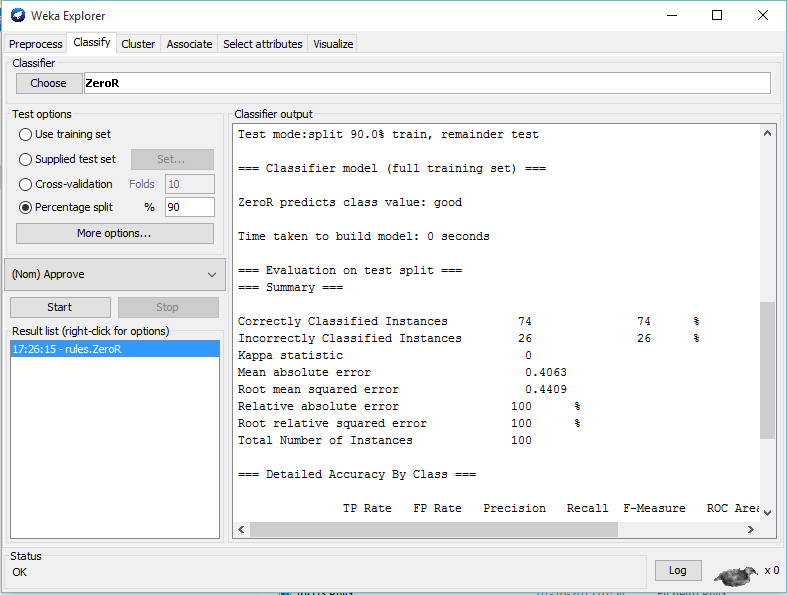


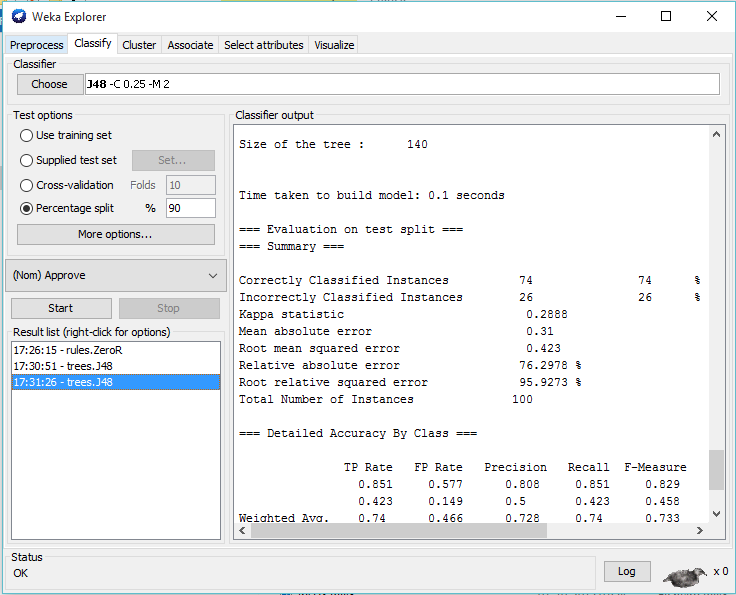
*Imagem reflectindo valores negativo sobre o atributo “Age” que é inusual.*

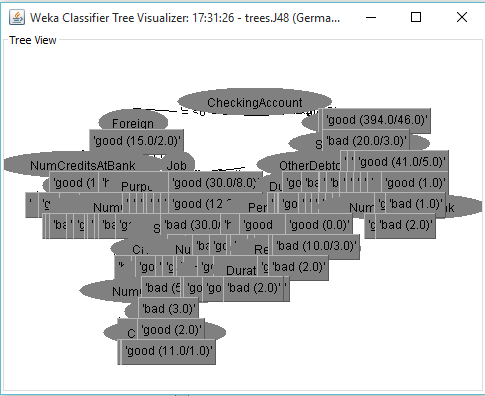
No ponto anterior, no ponto anterior encotrou-se várias instâncias que parecem estar corrompida, visto que alguns dos valores dos seus atributos não fazem sentido. Até um único ponto do gênero pode afectar significativamente o rendimento de um classificador.

Para remover o referido ponto do conjunto de dados utilizamos uma operação de pré-processamento (um filtro). Removemos todas as instâncias cujos valores de idade (atributo *age*) são inferiores a 0 anos. Na secção *Preprocess*, clicamos em *Choose* no painel de filtros. Seleccione *filters > unsupervised > instance > RemoveWithValues*. Clicamos no texto do filtro para alterar os seus parâmetros. Introduzimos 13 para o valor do índice do atributo (*attributeIndex*) e 0 para o ponto de corte (*splitPoint*). Clicamos *Ok* para definir os parâmetros e *Aplly* para aplicar o filtro aos dados. Visualizamos novamente os dados para verificar que o ponto de dados inválido foi removido e obtivemos o seguinte resultado.

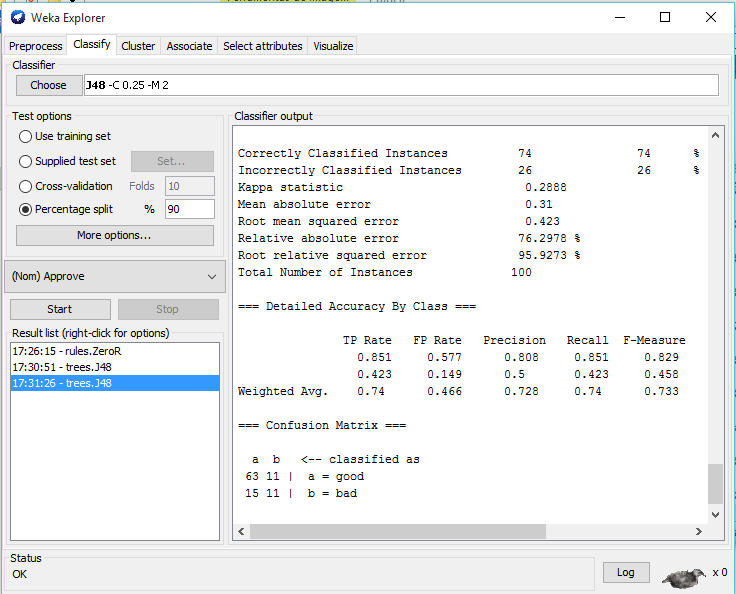


No painel de classificação, seleccione *Percentage split*, na secção *Test options*, e altere o seu valor para 90%. Com isto treinamos o classificador utilizando 90% dos dados e testamos o seu rendimento utilizando os restantes 10%, obtivemos o seguinte resultado.

Treinemos uma árvore de decisão. Para tal, seleccionamos *classifiers > trees > J48* e click *start*. J48 é a implementação no Weka do algoritmo C4.5 para a aprendizagem de árvores de decisão. Assim temos a seguinte imagem:

Uma vez treinado o classificador, a árvore aprendida é apresentada na janela da direita. A mesma também pode visualizar em forma gráfica fazemos click direito sobre o classificador, na lista de resultados à esquerda, e seleccionando *visualize tree*. Obtemos o seguinte resultado.

Observamos a saída do classificador e tratamos de responder algumas questões que nos foram colocadas:

* O rendimento do classificador é acima da média. A percentagem de instâncias correctamente classificadas é uma medida suficiente sim. Porque 69% das instâncias foram correctamente classificadas e 31% delas não. Analisando o número casos bem e mal classificados utilizando a matriz de confusão temos:

Das instâncias correctamente classificadas (74%) classificada 63 são boas e 11 são más, das incorrectamente classificadas (26%) 15 são boas e 11 são mas.

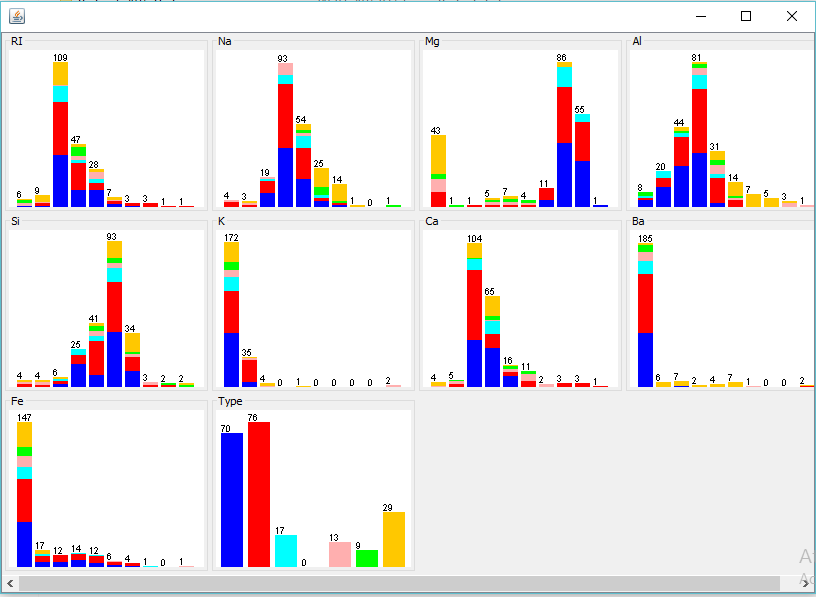
* Analise da árvore de decisão: As regras estabelecidas pela ávore de decisão não são razoaveis, Existem sim ramos parecendo absurdos.
* A árvore de decisão na classificação nos casos em que não existe nenhuma instância no conjunto de treino correspondente à determinada trajectória os nodos folhas são assinalados como (0.0).

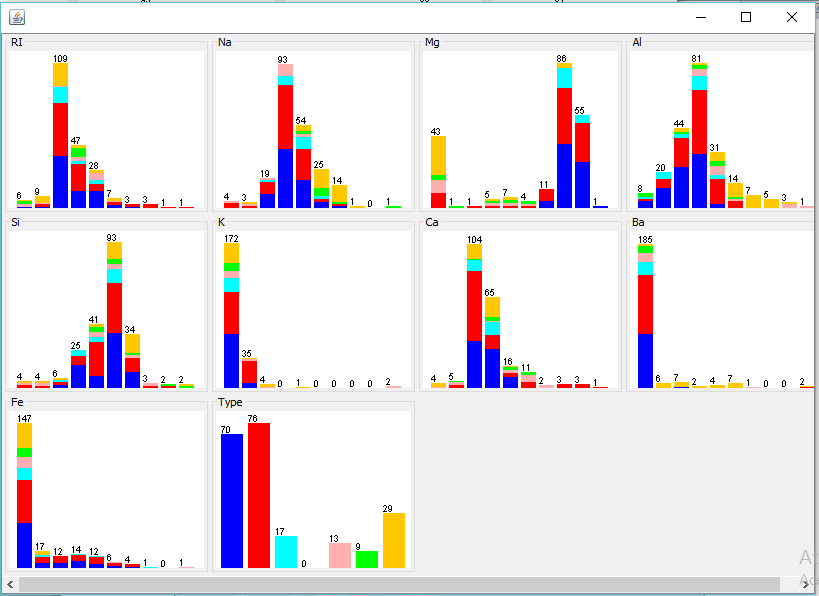
**Resposta da Tarefa 3:**

Nesta tarefa aplicaremos outra operação de pré-processamento que é a discretização de dados numéricos. Esta é uma operação utilizada nos casos em que temos conjuntos de dados com atributos numéricos e pretendemos converter os referidos valores à uma escala nominal.

Existem dois tipos de discretização: *não supervisionada*, não tem em conta as classes das instâncias, e *supervisionada*, tem em conta as classes das instâncias ao criar os intervalos. O principal método de *discretização não supervisionada* existente no Weka é *Weka.filters.unsupervised.attribute.Discretize*. O mesmo implementa duas variantes: *discretização com intervalos igual largura* (seleccionado por defeito) e com *intervalos de igual frequência*.

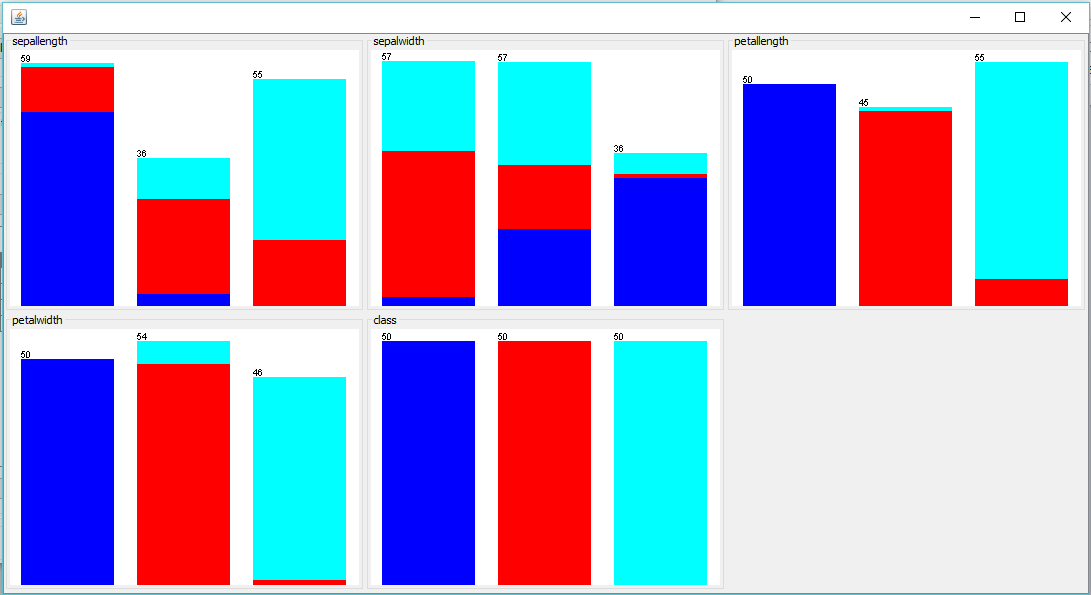
Carregamos o conjunto de dados *glass.arff* no explorador.

Aplicamos o método de *discretização não supervisionada* nos dois modos mencionados. Para tal no separador *Preprocess*, fez-se click no botão *Choose* e seleccionamos *filters > unsupervised > Discretize*. Obtivemos o seguinte resultado:

Para mudar de modo, fez-se click com o botão direito do rato na barra de edição diante do botão *Choose* e seleccionamos *Show properties* no menú de contexto. Na caixa de diálogo alteramos o valor da propriedade *useEqualFrequency de “False” para “True”*. Obtivemos o seguinte resultado:

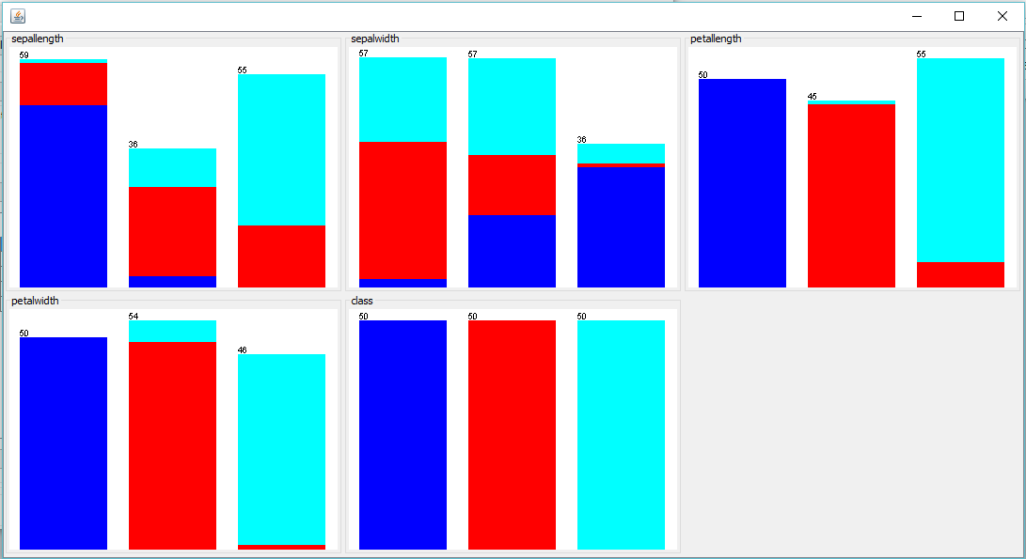
Observou-se o seguinte ao comparar os histogramas obtidos: Os histogramas correspondentes à *discretização com intervalos de igual frequência* **não** são bastante enviesados para **todos** os atributos, porque ao altermos o valor do atributo “*useEqualFrequency*” os atributos são experimentados na mesma frequência logo não há alterações significativa.

A principal técnica de *discretização supervisionada* de atributos numéricos no Weka é *weka.filters.supervised.attribute.Discretize*.

Carregamos o conjunto de dados *iris.arff*, aplicando a *discretização supervisionada* e observamos os histogramas obtidos. A *discretização supervisionada* criamos intervalos nos quais a distribuição das classes é consistente, ainda que a distribuição das instâncias varie de um intervalo para outro. Otivemos os seguintes gráficos:

Com base nos histogramas obtidos, o atributo discretizados que possui maior poder preditivo é o **sepalwidth.**

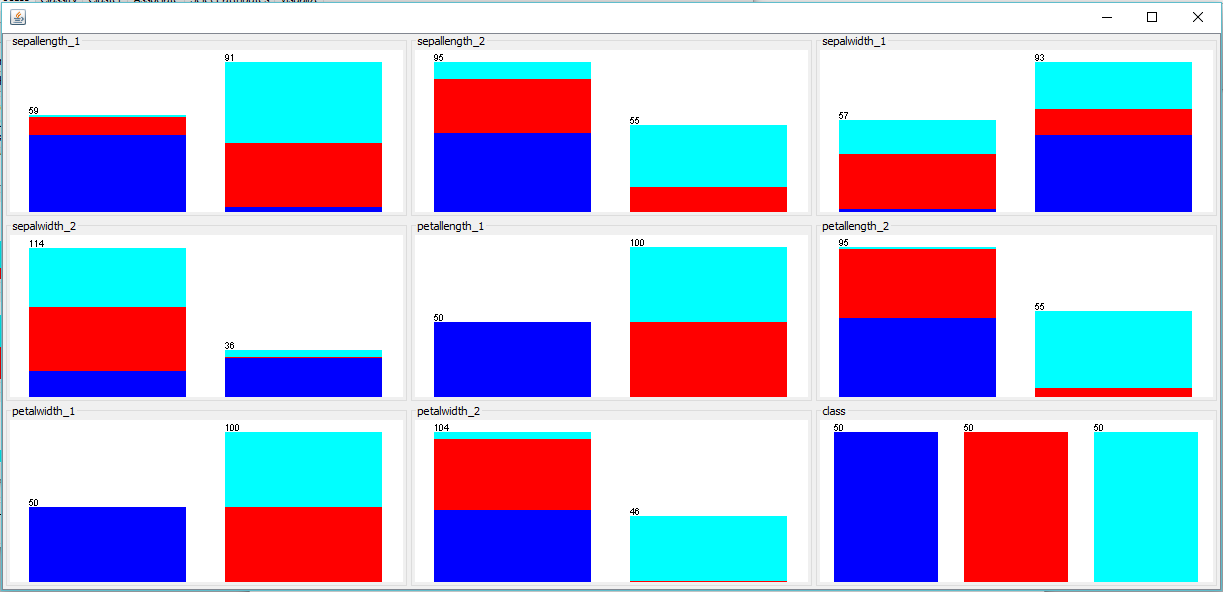
Carreguando novamente o conjunto *glass.arff* e aplicando a *discretização supervisionada*.



Para alguns atributos existe uma única barra no histograma isto significa que no referido atributo as instâncias possuem a mesma classififcação.

Os atributos discretizados são normalmente codificados como atributos nominais, com um único valor para cada rango. Ambos os filtros permitem a criação de atributos com valores binários em vez de multivalorados, activando a opção *makeBinary*, na caixa de diálogo de propriedades.

Escolhemos o conjunto de dados *“iris.arff”* e utlizando um dos métodos e usando-o para criar atributos binários. Comparando os resultados com as saídas produzidas quando a opção *makeBinary* é falsa. Observamos que os atributos binários são aqueles que possuem apenas duas correncias e que se o conjunto de dados estiver discretizado quase que não haverá diferencia quanto a opção *makeBinary* true ou false. Observou-se um único histograma para o conjunto de dados em questão.



Examinemos agora o efeito da discretização ao construir uma árvore de decisão com o algoritmo *C4.5* (designado *J48* no Weka) com os dados do conjunto *hionosphere.arff.* Este conjunto de dados contém informação sobre sinais de radar retornados da ionosfera. Consideram-se “Boas” instâncias aquelas que mostram evidências da existência de algum tipo de estrutura na ionosfera, enquanto que os sinais “Maus” são aqueles que atravessam directamente a ionosfera. Para mais informações, veja os comentários no ficheiro ARFF.

Iniciamos com a *discretização não supervisionada*.

Para o algoritmo *C4.5* (*J48*), *weka.classifiers.trees.J48*, comparamos a taxas de erro da validação cruzada e o tamanho das árvores geradas pelos (1) dados em bruto, (2) dados discretizados usando o método não supervisionado de discretização no modo por defeito e (3) dados discretizados pelo mesmo método com atributos binários.

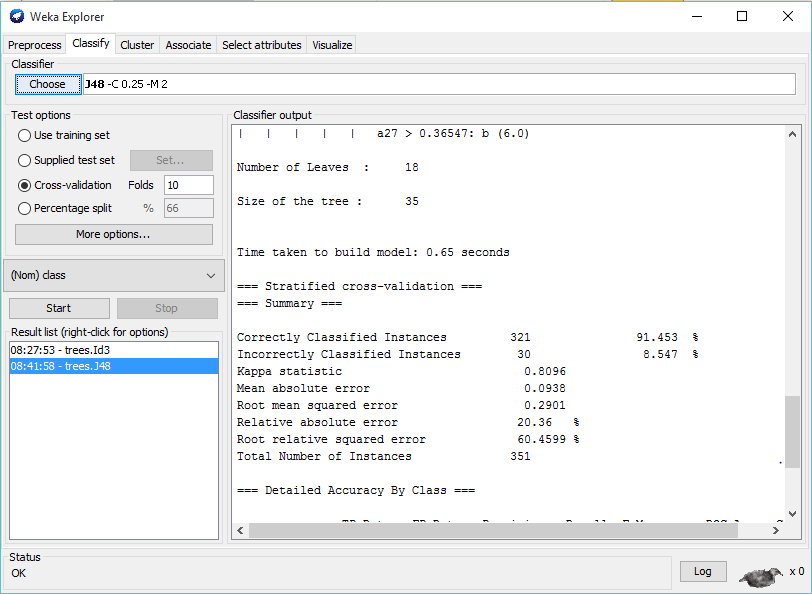
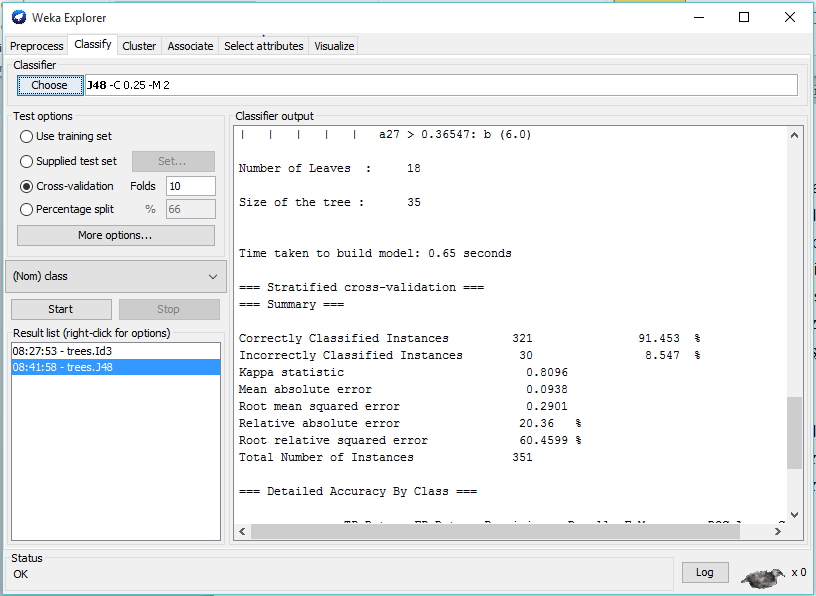


Imagem acima refletindo a taxas de erro da validação cruzada e o tamanho das árvores geradas pelos dados em bruto.



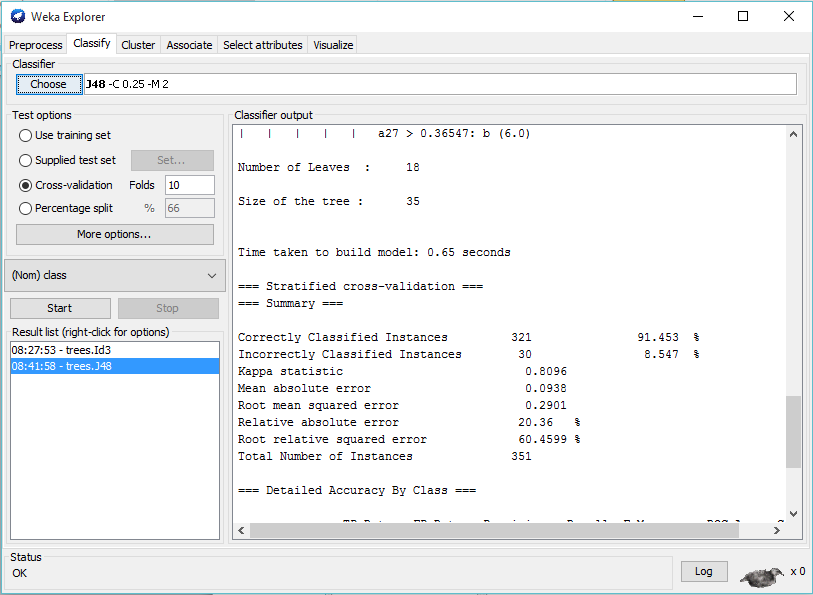
Imagem acima refletindo a taxas de erro da validação cruzada e o tamanho das árvores geradas pelos dados discretizados usando o método não supervisionado de discretização no modo por defeito.

Imagem acima refletindo a taxas de erro da validação cruzada e o tamanho das árvores geradas pelos dados discretizados usando dados discretizados pelo mesmo método com atributos binários.

Vejamos agora a *discretização supervisionada*. Neste caso, se utilizamos a totalidade dos dados para a discretização os resultados da validação cruzada são demasiadamente optimistas. O procedimento correcto é costruir o filtro utilizando apenas os dados de treino e avaliar o classificador sobre os dados de teste, utilizando os intervalos de discretização determinados com os dados de treino. Para tal se utiliza um *metaclassificador* incluído no Weka, *FilteredClassifier* (*weka.classifiers.meta. FilteredClassifier*).

Utilizando *FilteredClassifier* e *J48*, comparando a taxa de erro da validação cruzada e o tamanho das árvores geradas pela (4) *discretização supervisionada*, no modo definido por defeito, e (5) *discretização supervisionada* com atributos binários.

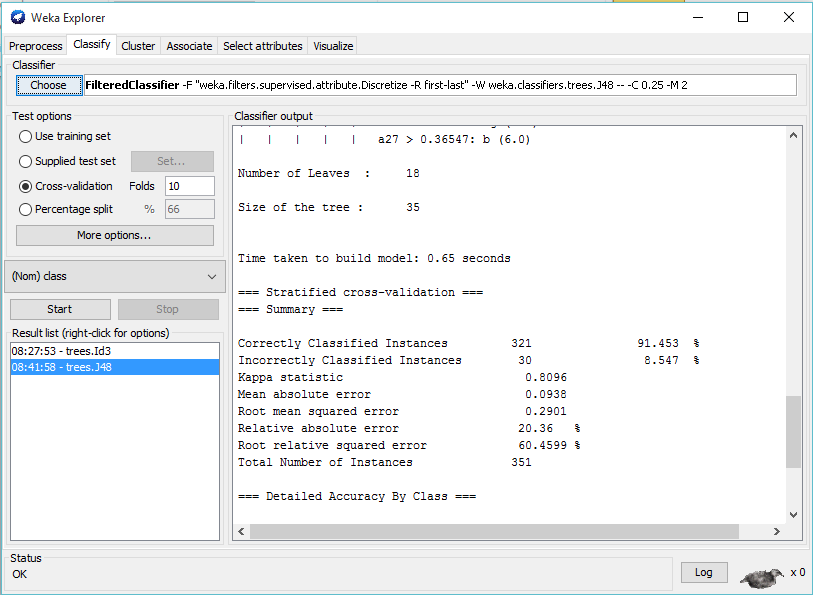


Imagem acima refletindo a taxa de erro da validação cruzada e o tamanho das árvores geradas pela *discretização supervisionada*, no modo definido por defeito.

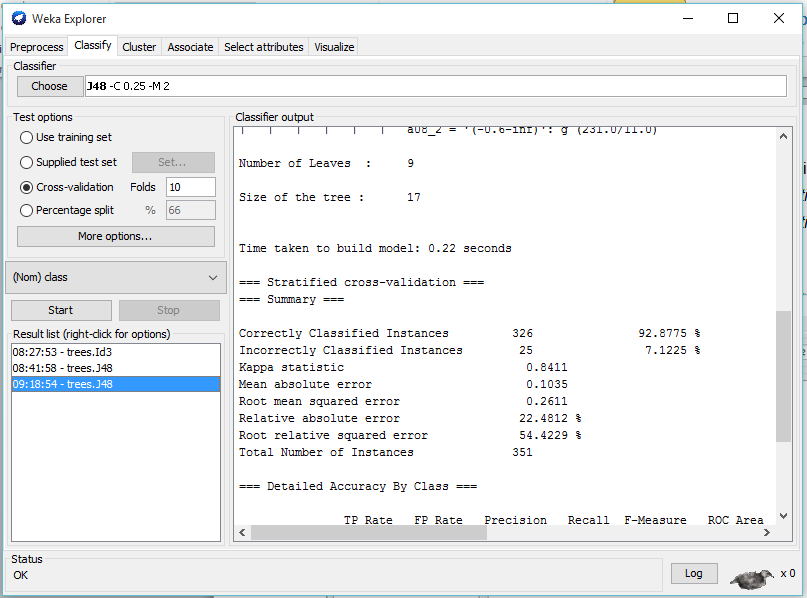


Imagem acima refletindo a taxa de erro da validação cruzada e o tamanho das árvores geradas pela *discretização supervisionada* com atributos binários.

Em função da nossa análise constatou-se que nesse caso não houve diferença entre a árvore de decisão gerada apartir dos dados brutos e a árvore de decisão gerada por dados discretizados.