



Universidade do Minho
Escola de Engenharia

Mineração de Dados

Mini Teste 1 - Individual
MiEI - 4º Ano - 1º Semestre

A85308 Filipe Miguel Teixeira Freitas Guimarães

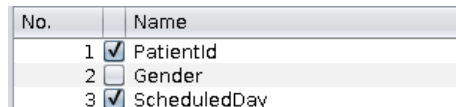
Braga,
18 de novembro de 2020

1 Pergunta 1

Considere o dataset CBrasil. Pretende-se elaborar um estudo para o desenvolvimento de um modelo de previsão para analisar as faltas às consultas nos vários centros de saúde do Rio de Janeiro. O objetivo é ter um estudo sobre possíveis modelos pra prever/identificar os pacientes que tem tendência a faltar às consultas marcadas. Temos de pré-processar os dados por forma a:

- **Eliminar atributos redundantes (ou com pouco valor informativo)**

Para avaliar que dados são irrelevantes decidi analisar que dados teriam menos relação com a previsão a ser estudada. Os dados que não têm qualquer valor são os *PatientId* e o *SheduleDay*.



No.	Name
1	<input checked="" type="checkbox"/> PatientId
2	<input type="checkbox"/> Gender
3	<input checked="" type="checkbox"/> ScheduledDay

Figura 1: Remover atributos redundantes

- **Forçar atributos booleanos ou categóricos a serem mesmo booleanos ou categóricos (e não interpretar como numéricos como estão nos dados!)**

Para forçar os dados a terem a correta atribuição recorri ao *Weka* e usei o filtro *NumericToNominal*, precisando só de alterar de novo a idade para numeric no meu editor de texto.

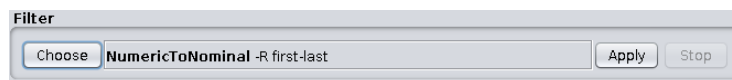


Figura 2: Remover atributos redundantes

Depois de atingir um dataset com os elementos relevantes apresente resultados que permitam responder às seguintes questões:

1.1 Qual os atributos a considerar para construir o modelo de previsão? Justifique.

Com tudo que já disse, os atributos a considerar serão então o *Gender*, *AppointmentDay*, *Age*, *Neighbourhood*, *Shcholarship*, *Hipertension*, *Diabetes*, *Alcoholism*, *Handcap*, *SMS,received* e *No-show* sendo que contibuem em conjunto para prever o facto da falta ou não nas consultas. Em relação ao *AppointmentDay* que podia ser um atributo a retirar achei, por não ter tantos diferentes, seria uma boa opção a considerar e com testes feitos verifiquei que o erro aumenta quando retiro este atributo.

1.2 Qual atributo com maior valor informativo?

Como podemos observar pela árvore gerada pelo algoritmo *J48* sabemos que, teoricamente, o atributo com maior valor informativo será o *AppointmentDay*.

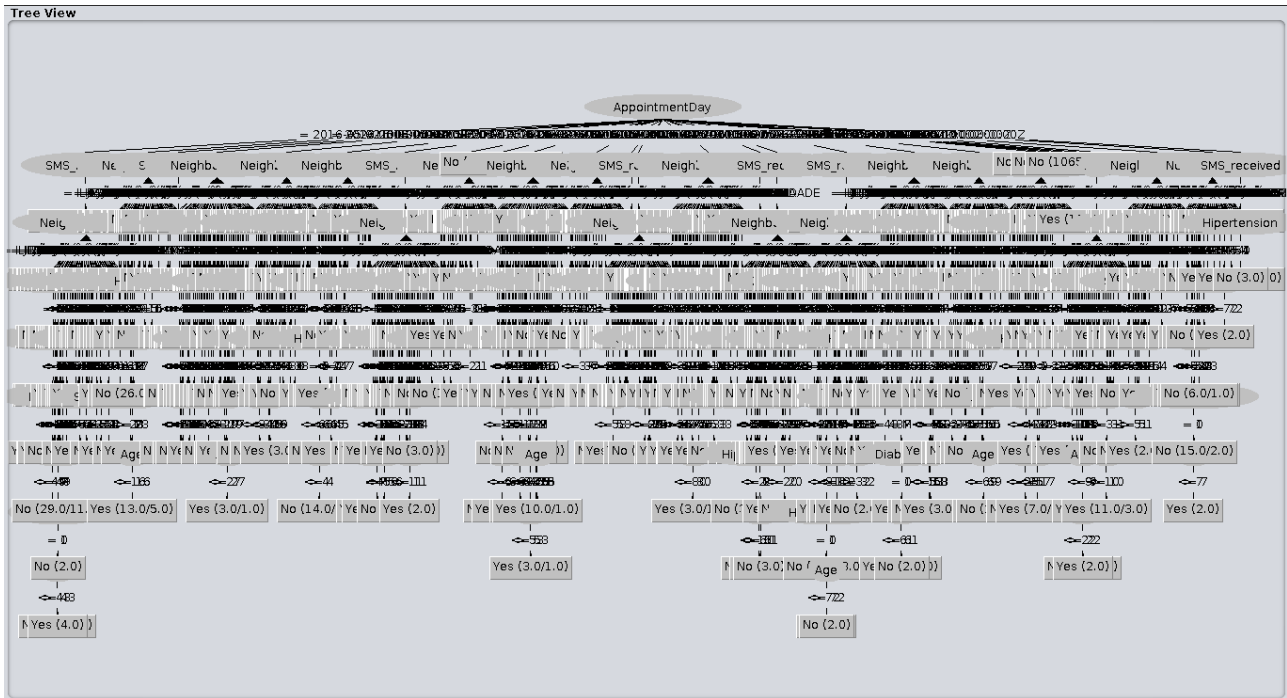


Figura 3: Árvore gerada pelo *J48*

Mas isto não fará muito sentido achando por intuição ser o *Neighbourhood* que terá o maior valor informativo.

2 Pergunta 2

Use o *WEKA* e as suas implementações de *NaiveBayes*, *BayesNet* e *J48* para responder às seguintes perguntas (apresente resultados obtidos por validação cruzada e variação de hiper-parâmetros dos três algoritmos):

2.1 Qual o modelo que escolhia para implementar em tempo real dentro destes 3 (e suas variantes)? Justifique.

Apoiado nos testes demonstrados nas figuras seguintes, e também nas figuras das perguntas seguintes o modelo que escolhia para implementar seria o *NaiveBayes* com discretização. É o que tem melhor resultados comparando com o tempo que demora. Como o *J48* demora bastante mais será uma escolha secundária mesmo tendo melhores resultados.

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      36347      62.5551 %
Incorrectly Classified Instances    21757      37.4449 %
Kappa statistic                     0.1659
Mean absolute error                 0.4554
Root mean squared error             0.4776
Relative absolute error             94.4538 %
Root relative squared error         97.2737 %
Total Number of Instances          58104

```

Figura 4: *NaiveBays* sem discretização

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      36376      62.605 %
Incorrectly Classified Instances    21728      37.395 %
Kappa statistic                     0.1669
Mean absolute error                 0.4554
Root mean squared error             0.4776
Relative absolute error             94.4524 %
Root relative squared error         97.2643 %
Total Number of Instances          58104

```

Figura 5: *NaiveBays* Com discretização

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      36376      62.605 %
Incorrectly Classified Instances    21728      37.395 %
Kappa statistic                     0.1669
Mean absolute error                 0.4554
Root mean squared error             0.4776
Relative absolute error             94.4482 %
Root relative squared error         97.265 %
Total Number of Instances          58104

```

Figura 6: *BaysNet*

```

Time taken to build model: 2.08 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      37686           64.8596 %
Incorrectly Classified Instances    20418           35.1404 %
Kappa statistic                    0.2386
Mean absolute error                 0.4319
Root mean squared error             0.4892
Relative absolute error             89.5776 %
Root relative squared error         99.6271 %
Total Number of Instances          58104

```

Figura 7: *J48*

2.2 Em termos de classe *no-show=yes* qual o melhor modelo? Justifique?

Para analisar melhor os algoritmos com as diferentes opções decidi compilar os resultados com no-show=yes na seguinte tabela

Algoritmo	Método de Procura	Descretização	LaPlace	Pruning	Tempo (/fold)	Precision	Recall	F-Measure	Roc Area
NaiveBayes	NA	Off	NA	NA	0.12	0.568	0.568	0.411	0.411
NaiveBayes	NA	On	NA	NA	0.18	0.569	0.322	0.411	0.637
BayesNet	Simulated Annealing	NA	NA	NA	17.35	0.614	0.304	0.407	0.660
BayesNet	K2	NA	NA	NA	0.16	0.569	0.322	0.411	0.637
BayesNet	Hill Climber	NA	NA	NA	0.27	0.571	0.324	0.413	0.638
J48	NA	NA	On	On	1.98	0.591	0.432	0.499	0.648
J48	NA	NA	Off	On	2.02	0.591	0.432	0.499	0.639
J48	NA	NA	Off	Off	2.06	0.535	0.467	0.499	0.617

Como se pode verificar o que terá mais precisão, entre outros fatores, será o *BayesNet* recorrendo ao método de procura *Simulated Annealing* mas também é o que leva mais tempo.

2.3 Para o modelo derivado do algoritmo J48 mostra as várias árvores possíveis de obter por diferentes configurações de pruning. Tente explicar os vários desempenhos (dos vários modelos derivados).

Mais uma vez decidi compilar todos os testes numa tabela para ser mais fácil de analisar.

Valor de Confiança	Prunning	Correctly Cllsified Instances	No Show	Precision	Recall	F-Measured	Roc Area
0.25	Off	61.9338 %	Yes	0.535	0.467	0.499	0.617
			No	0.665	0.723	0.693	0.617
0.01	On	63.9388 %	Yes	0.630	0.269	0.377	0.637
			No	0.641	0.892	0.746	0.637
0.15	On	64.8114 %	Yes	0.624	0.333	0.434	0.640
			No	0.655	0.863	0.745	0.640
0.25	On	64.8596 %	Yes	0.591	0.432	0.499	0.639
			No	0.673	0.796	0.729	0.639
0.4	On	63.8717 %	Yes	0.569	0.449	0.502	0.625
			No	0.671	0.768	0.717	0.625
0.5	On	62.6188 %	Yes	0.547	0.459	0.499	0.620
			No	0.667	0.740	0.702	0.620

O *prunning* divide-se em duas estratégias diferentes:

- O *pré-pruning* que consiste em para a expansão de um ramo que o informação se torna pouco fiável, ocorrendo *underfitting* quando para demasiado cedo.
- O *pós-pruning* que consiste em fazer crescer a árvore até ao final e só depois retirar os ramos com informação pouco fiável.

Decidi testar então, o que considere mais relevante (como se pode ver na tabela), ou seja, sem *pruning* com confiança de 0.25 e com *pruning* com diferentes valores de confiança.

Consegui verificar que os resultados melhoram com o aumento do valor de confiança até aos 0.25, baixando depois disso. Verifico assim que antes de 0.25 ocorre *underfitting* e depois disso *overfitting*.