**Relatório Atividade Final - Opção 2**

Alunos:

[Eduardo Wanderley de Siqueira Araujo](mailto:ewsa@cin.ufpe.br)

Jefferson Lovis

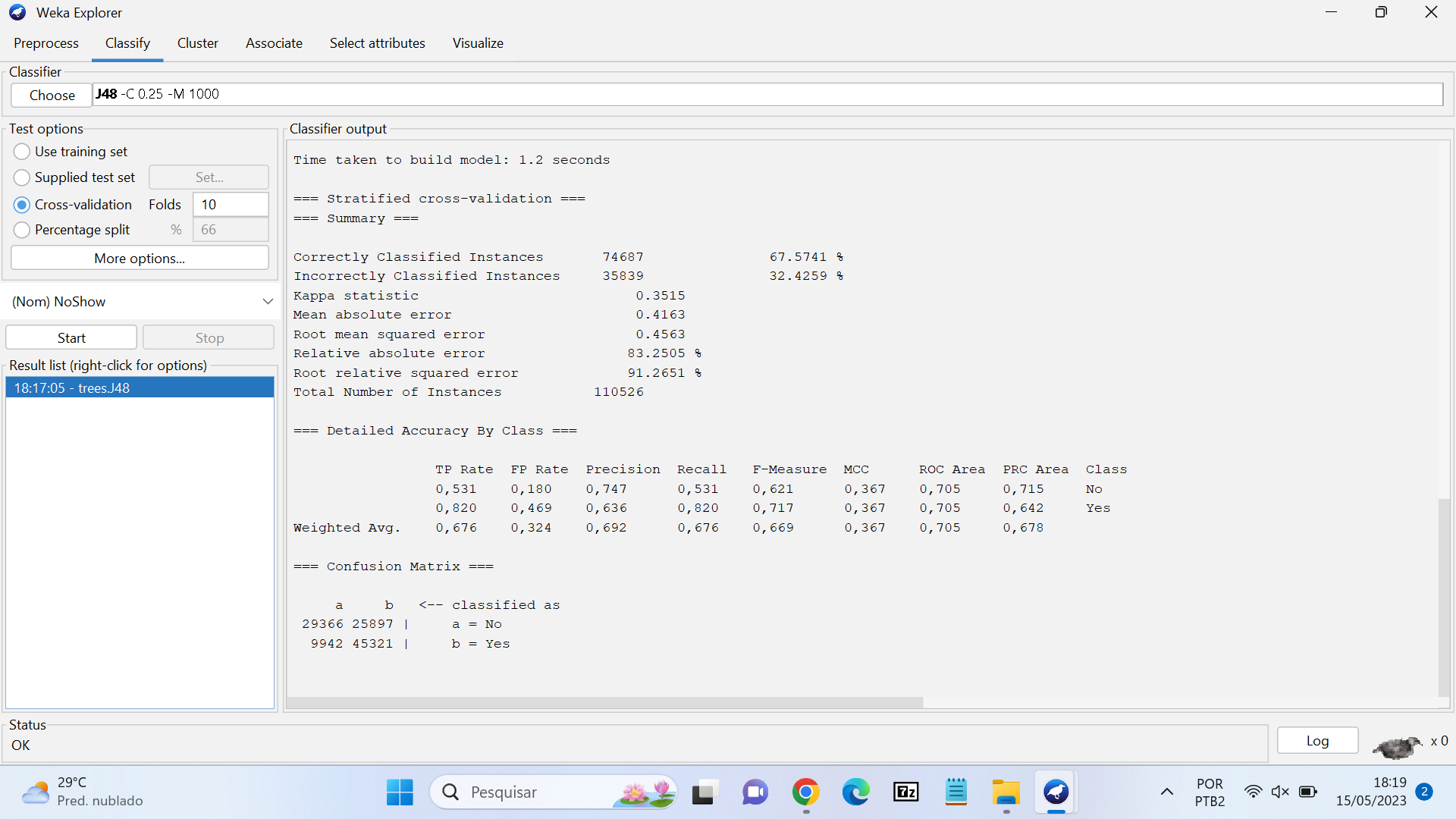
Ricardo César Montezuma

1.Questão

Passo 1:

Utilizamos o filtro Resample. Alteramos em sua configuração: biasToUniformClass para 1.0. Tornando a quantidade de instâncias de cada classe iguais

Passo 2: Fomos para a aba Classify e escolhemos o algoritmo classificador como J48, utilizando o mínimo número de objetos (minNumObj) igual a 1000. O resultado está representado abaixo pela figura.

passo 3: No modelo base, utilizando todos os atributos, obtivemos uma curva ROC de 0,705 e acurácia de 67.5741%. Existem algumas regras que apresentam boa cobertura e confiança para esse caso. Para identificar essas regras, fomos cortando aquelas que não tem correlação e aparentam não possuir causa e efeito. Além das que tem uma cobertura baixa e ocorrem poucas vezes nas 110526 instâncias totais. Segue abaixo algumas regras:

Delay <= 5.616389: No (27615.0/4490.0).

(Rubem) Confiança: 83,74% → (27615 - 4490) / (27615). O que indica que 23125 instâncias compareceram a consulta

(chatGPT) Confiança = 86.07% → 27615.0 / (27615.0 + 4490.0)

Esse valor garante uma boa confiança na regra, porém como podemos observar abaixo, a cobertura é de apenas 29,27%, sendo apenas razoável e não acontecendo com tanta frequência.

(Rubem, Ricardo) Cobertura: 24,98% → 27615/110526

(chatGPT) Cobertura = 29.27% → (27615 + 4490) / 110526

A confiança de 86,07% significa que, quando a variável "Delay" é menor ou igual a 5.616389, o modelo J48 classificou corretamente cerca de 86.07% das instâncias como "No".Já a cobertura dessa regra específica é de 29.27% indicando se aplica a cerca de 29.27% das instâncias no conjunto de dados total. Logo, apesar de uma ótima confiança, a cobertura é apenas razoável.

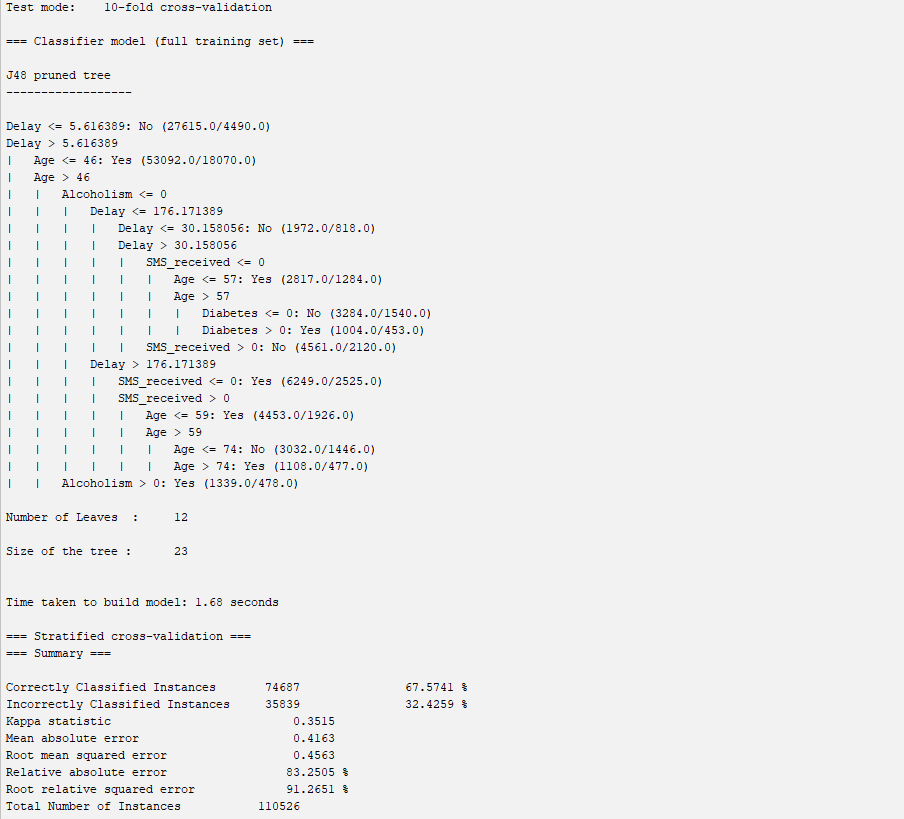
Delay > 5.616389 AND Age <= 46: Yes (53092.0/18070.0)

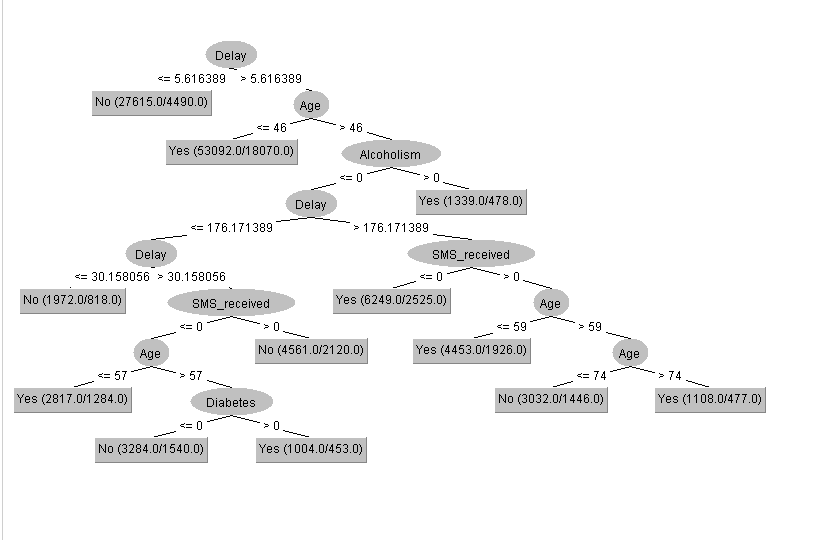
Confiança = 74.62% → 53092.0 / (53092.0 + 18070.0)

Cobertura = 65.95% → (53092 + 18070) / 110526

A confiança de 74.62% significa que, quando a variável "Delay" é maior que 5.616389 e Age é menor que 46, o modelo J48 classificou corretamente cerca de 74.62% das instâncias como "No".Já a cobertura dessa regra é de 65.95% a um bom número de instâncias no conjunto de dados total. Sendo uma regra importante de ser observada.

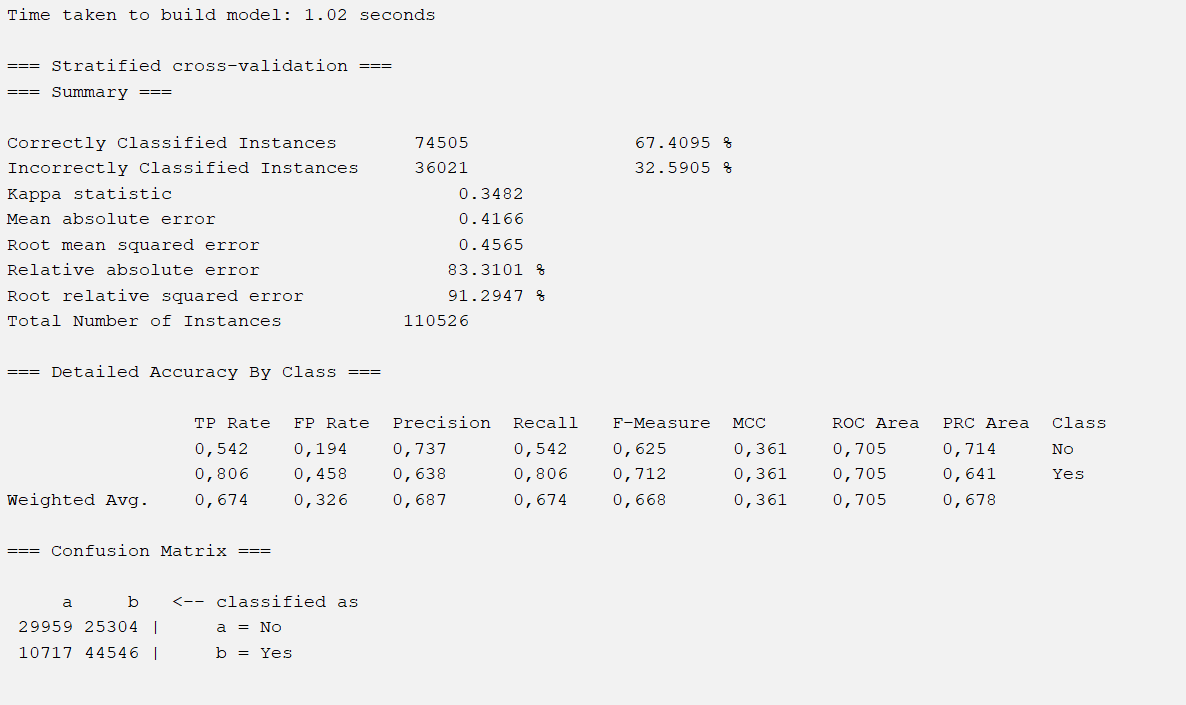
As duas regras vieram da árvore de decisão, representadas nas figuras abaixo.

Todas as outras regras apresentadas nesse caso, possuem cobertura menor que 10% e por isso ocorrem com pouca frequência ou apresentam atributos irrelevantes em suas regras, não valendo a pena serem mencionadas.

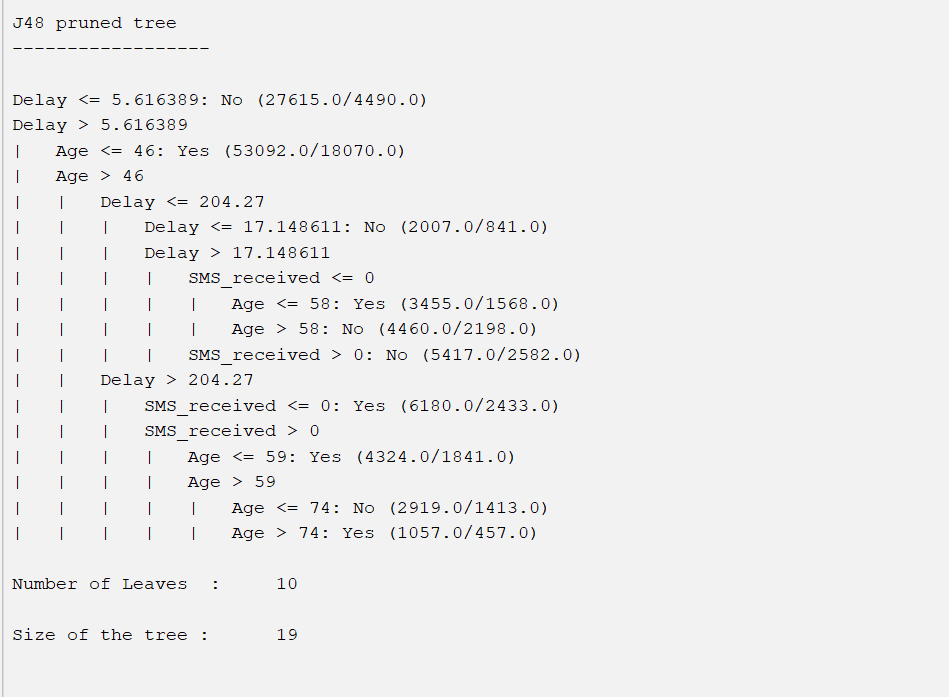


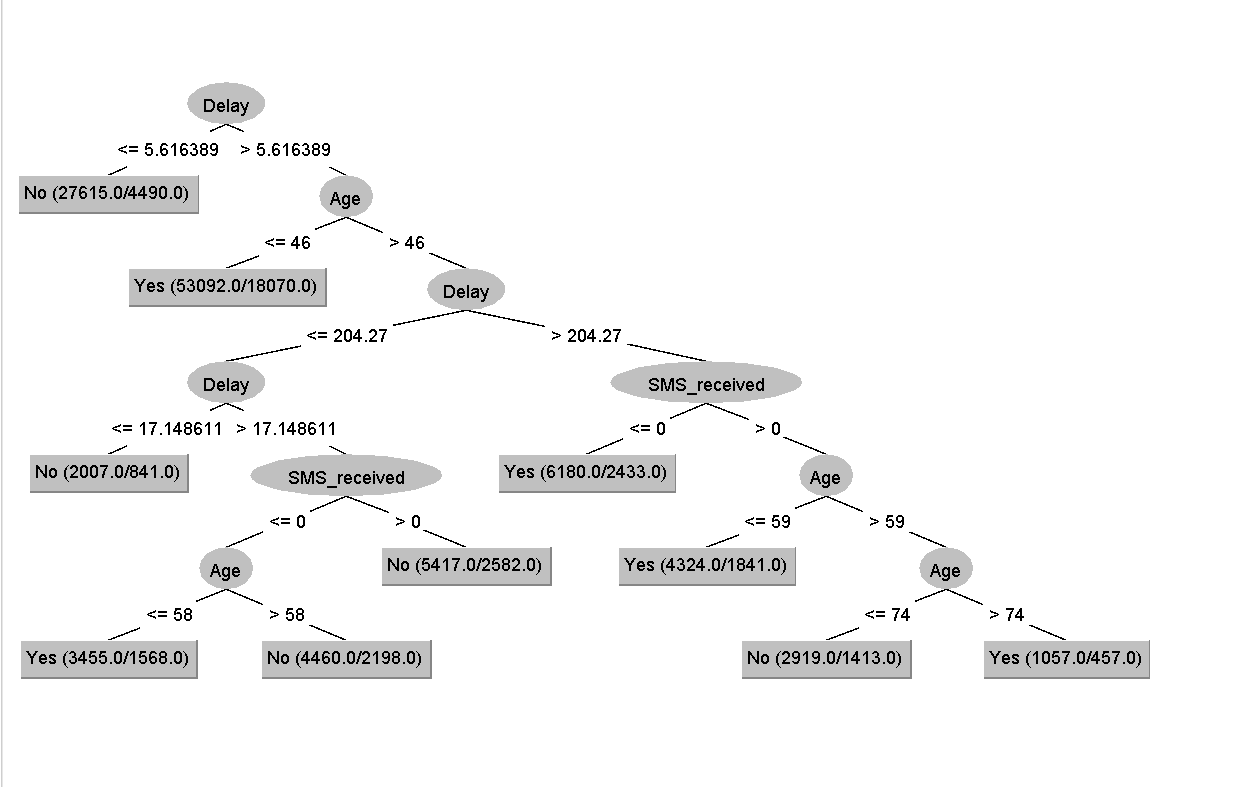
Teste 1

Removendo os atributos Scholarship, Hipertension e Diabetes, e utilizando as mesmas configurações do caso anterior, obtivemos a seguinte resposta e árvore de decisão:



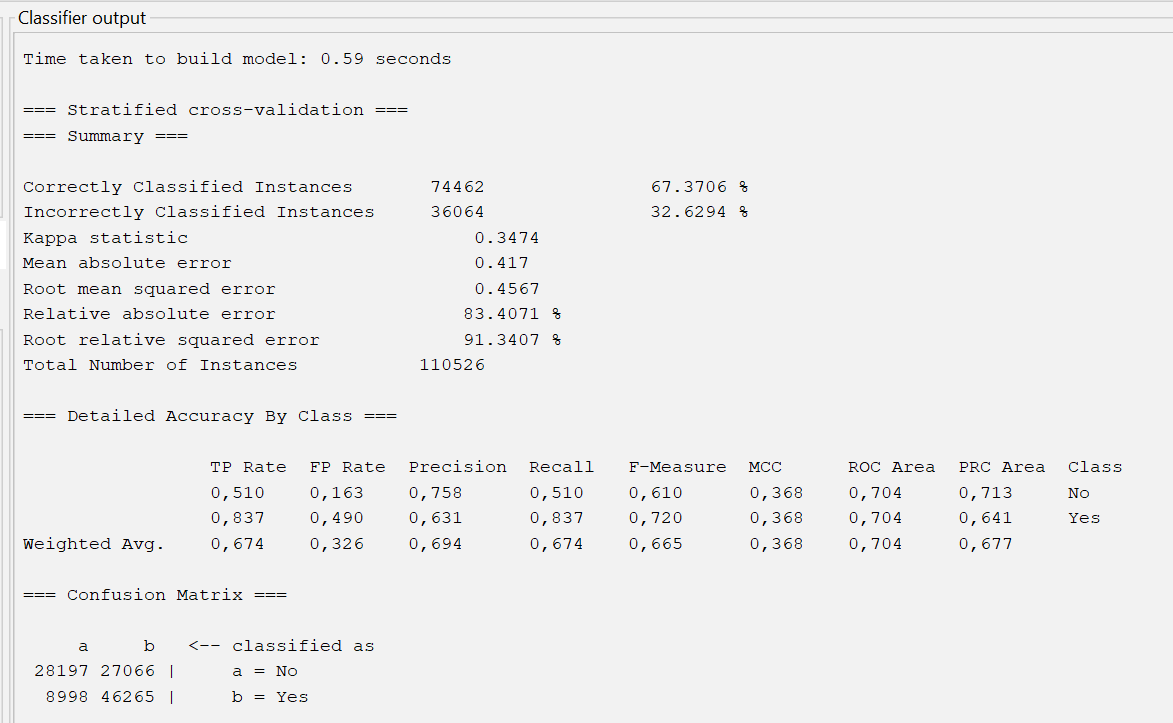
A acurácia da resposta diminui muito pouco, para 67.4095% e a curva ROC se manteve com 0,705. Boas regras que podem ser tiradas desse teste 1 também são iguais ao caso anterior, como podemos ver na árvore de decisão nas figuras abaixo.

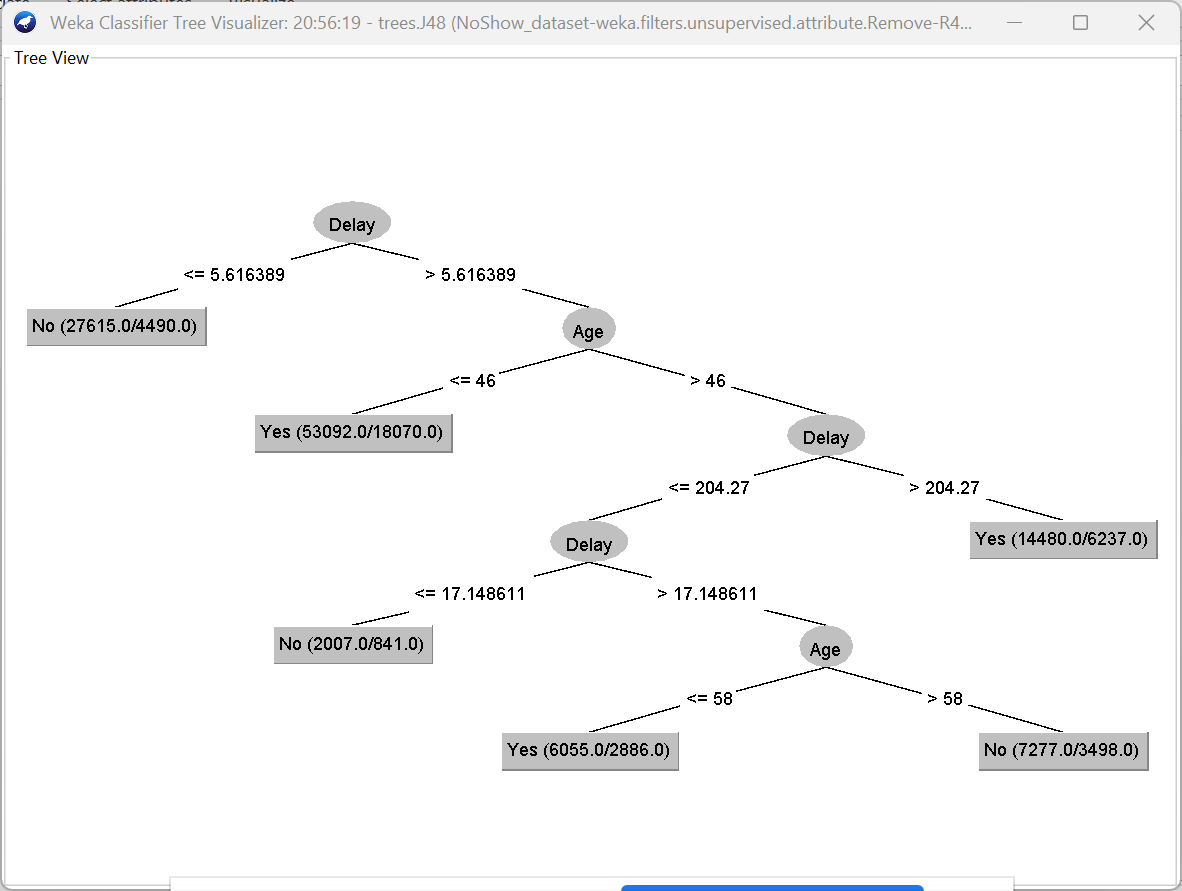




Teste 2

Removendo os atributos Gender, Hipertension e Diabetes, Alcoholism, Scholarship, Handicap and SMS\_received obtivemos a seguinte resposta e árvore de decisão:



A acurácia da resposta diminui um pouco mais, para 67.3706% e a curva diminuiu para 0,704. Podemos identificar uma regra razoável desse teste 2, conforme podemos observar na árvore de decisão abaixo:

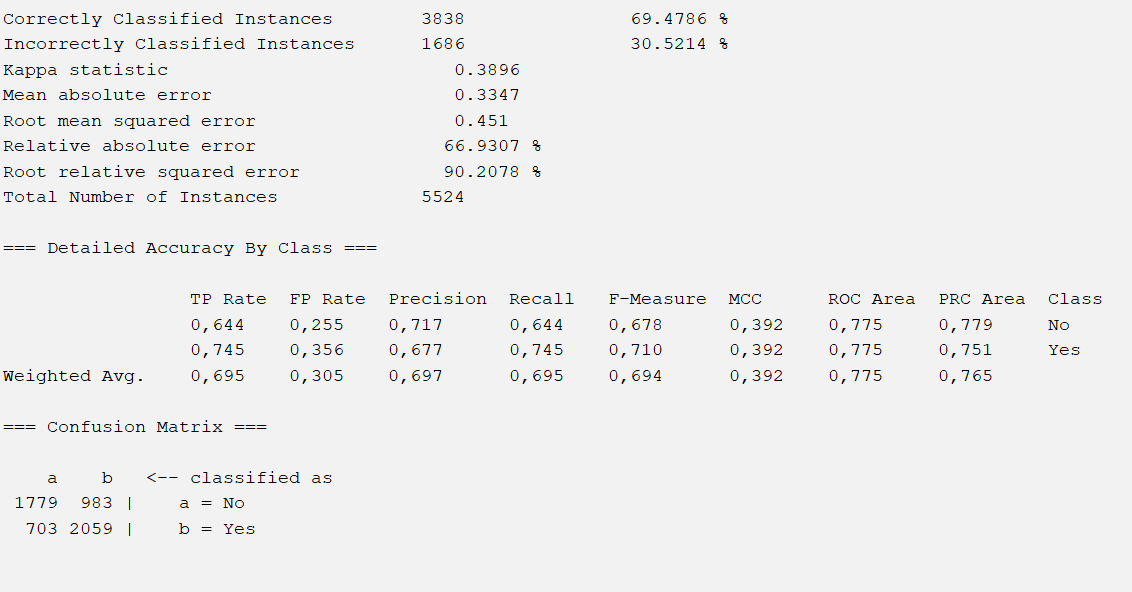
Delay > 5.616389 AND Age > 46 AND Delay > 204.27: Yes (14480.0/6237.0)

* Confiança: 69.86% → 14480.0 / (14480.0 + 6237.0)
* Cobertura: 18.74% → (14480 + 6237) / 110526

A confiança dessa regra é de 69.86%. Isso significa que, quando a variável "Delay" é maior que 5.616389, e a variável "Age" é maior que 46 e a variável "Delay" é maior que 204.27, o modelo J48 classificou corretamente cerca de 69.86% das instâncias como "Yes". Com a cobertura apresentada acima, vemos que essa regra se aplica a cerca de 18.74% das instâncias no conjunto de dados total, não tendo uma cobertura tão boa, e também com uma confiança menor que as duas regras já apresentadas, ela acaba sendo a pior das três.

Questão 2

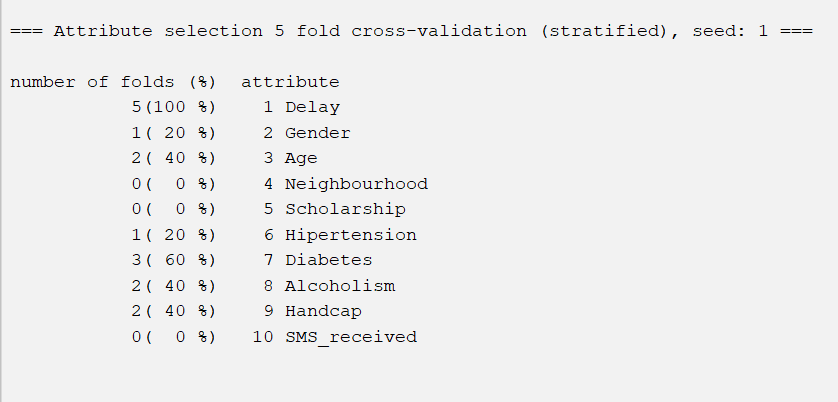
O algoritmo IBK foi aplicado ao dataset reduzido a 5% disponibilizado para atividade , com todos os atributos iniciais e utilizando o resample com o fator biasToUniformClass para 1.0, Tornando a quantidade de instâncias de cada classe iguais. com o KNN de 5, obtendo os seguintes resultados:



A quantidade de instâncias classificadas corretamente foi de aproximadamente 69.48% do total de instâncias e a área sob a curva ROC foi de 0,775.Já a precisão, uma métrica que indica a proporção de instâncias classificadas corretamente como positivas (TP) em relação ao total de instâncias classificadas como positivas (TP + FP), também aumentou para o valor de 0,697. Para tentar melhorar o resultado, utilizamos a técnica de select attributes com as seguintes configurações:

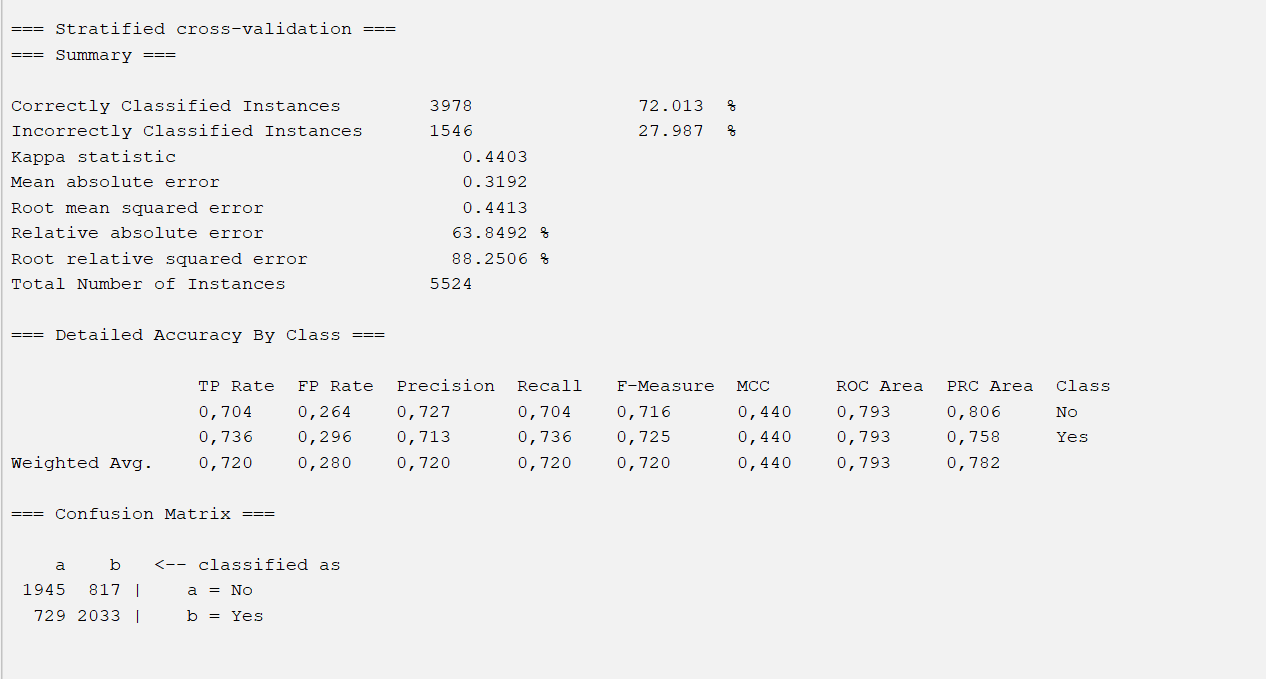
* WrapperSubsetEval: Classifier = IBK com knn 5, evaluationMeasure = AUC(ROC curve) e folds = 2
* GreedyStepWise: searchBackwards = False
* Cross-Validation: 5 folds em Attribute selection mode

Obtendo o seguinte resultado para cada atributo:



Teste 1

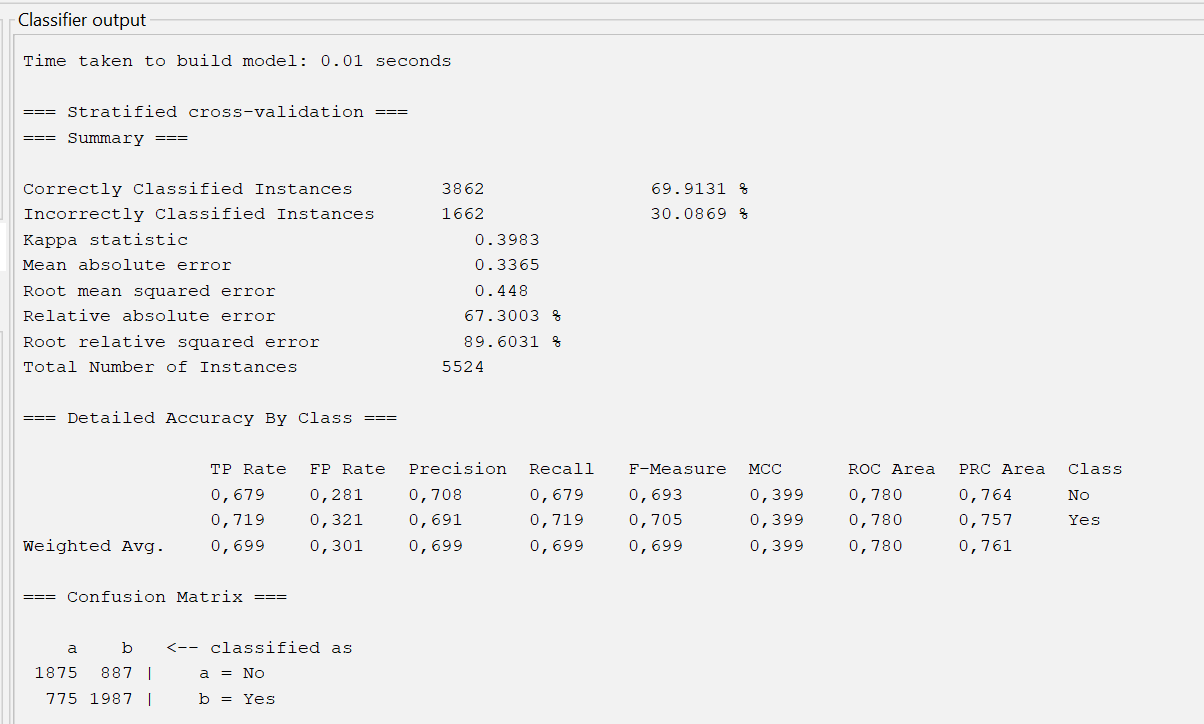
Retirando os atributos iguais ou abaixo de 20%, Gender, Age, Neighbourhood, Scholarship, Hipertension e SMS\_received, pois foram identificados como possivelmente dispensáveis graças ao baixo percentual apresentado, (apenas presentes em uma das 5 folhas \*\*Eu acho), rodamos novamente o modelo IBK com as mesmas configurações anteriores e obtivemos:



Removendo os atributos citados, conseguimos melhorar o desempenho da área sob a curva ROC para 0,793 e o número de instâncias classificadas corretamente aumentou para 72.013%, aumentando a proporção de instâncias classificadas corretamente em relação ao total delas. Já a precisão, também aumentou para o valor de 0,720.

Teste 2

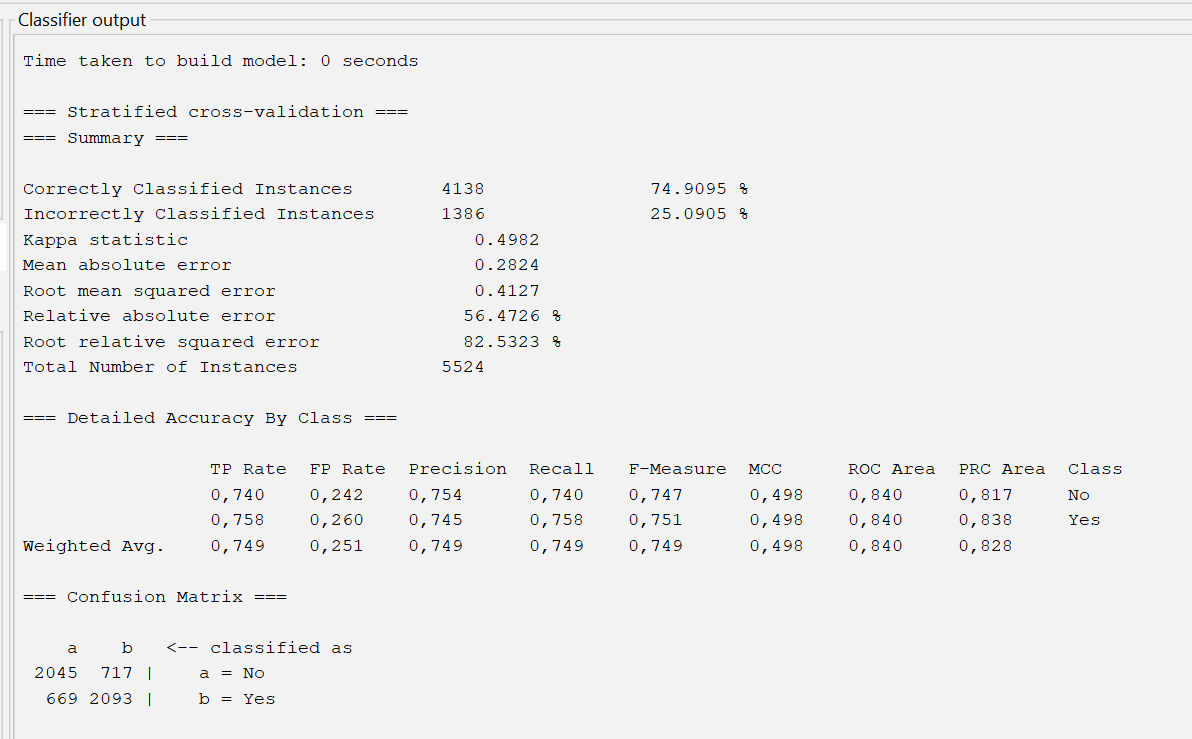
Removendo os atributos Gender, Age, Neighbourhood, Scholarship, Hipertension e SMS\_received utilizando novamente o modelo IBK com as mesmas configurações anteriores obtivemos o seguinte resultado:



Observamos uma leve queda nas taxas de desempenho, a área sob a curva ROC ficou em 0,780 o número de instâncias classificadas corretamente ficou em 69.9131% e a precisão em 0,699.

Teste 3

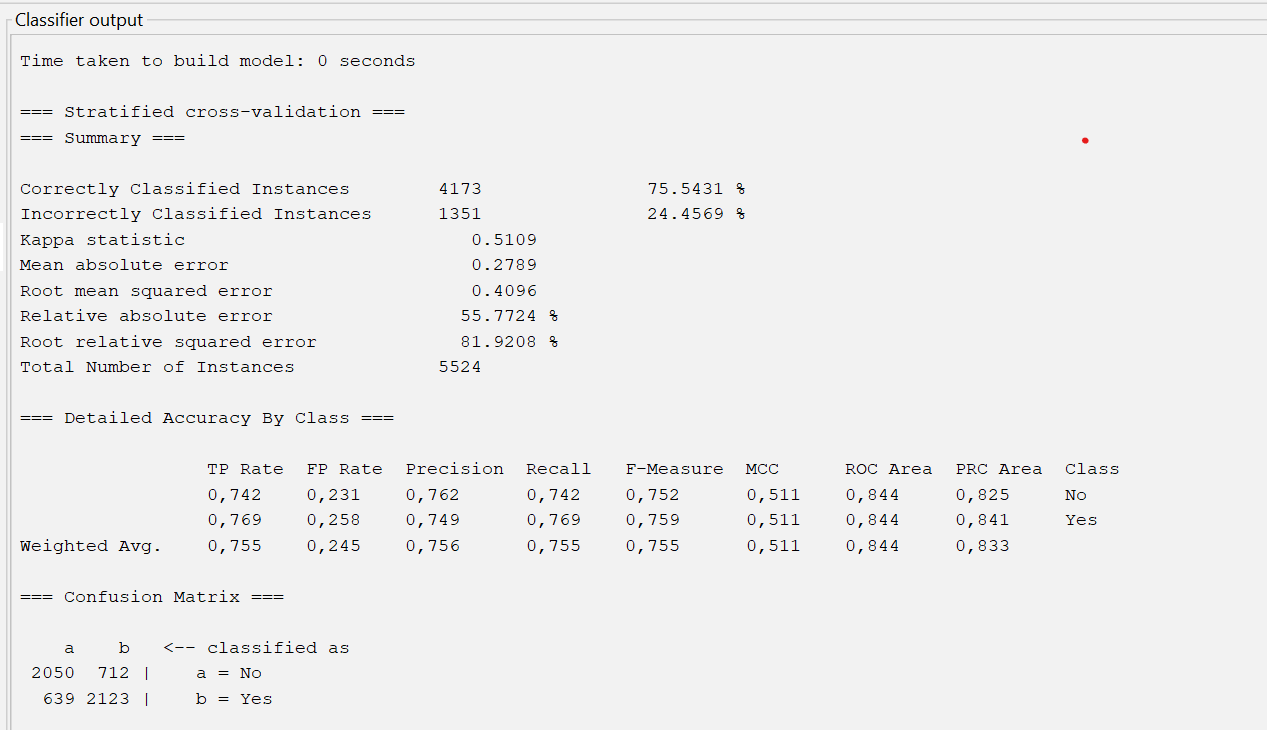
Removendo os atributos Gender, Age, Neighbourhood, Scholarship, Hipertension, Handcap e SMS\_received utilizando novamente o modelo IBK com as mesmas configurações anteriores obtivemos o seguinte resultado:



Observamos uma melhora no desempenho da área sob a curva ROC para 0,840 e o número de instâncias classificadas corretamente aumentou para 74.9095%, aumentando a proporção de instâncias classificadas corretamente em relação ao total delas. Já a precisão, também aumentou para o valor de 0,749.

Teste 4

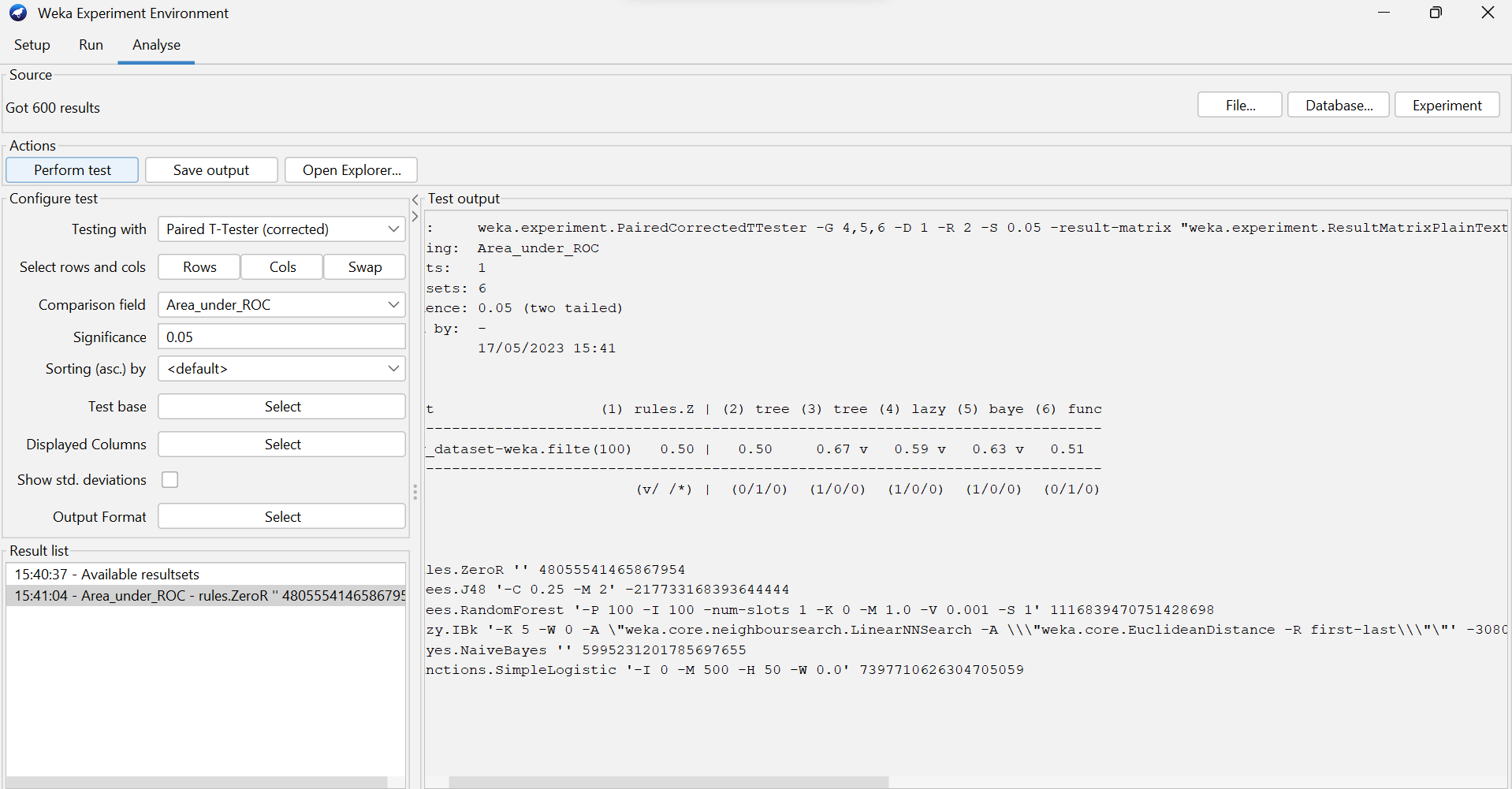
Removendo os atributos Gender, Age, Neighbourhood, Scholarship, Hipertension, Alcoholism e SMS\_received utilizando novamente o modelo IBK com as mesmas configurações anteriores obtivemos o seguinte resultado:



Observamos uma melhora no desempenho da área sob a curva ROC para 0,844 e o número de instâncias classificadas corretamente aumentou para 75.5431%, aumentando a proporção de instâncias classificadas corretamente em relação ao total delas. Já a precisão, também aumentou para o valor de 0,756.

Questão 3

3. Usando o ambiente Experimenter do WEKA, realize experimentos de validação cruzada com os algoritmos ZeroR, Regressão Logística, IBk, J48, Naive Bayes e Random Forest.



Dataset (1) rules.Z | (2) tree (3) tree (4) lazy (5) baye (6) func

------------------------------------------------------------------------------------

NoShow\_dataset-weka.filter(100) 0.50 | 0.50 0.67 v 0.59 v 0.63 v 0.51

------------------------------------------------------------------------------------

(v/ /\*) | (0/1/0) (1/0/0) (1/0/0) (1/0/0) (0/1/0)

1. zeroR
2. J48,
3. RandomForest
4. IBk com kNN=5,
5. NaiveBayes
6. SimplesLogistic

Alteramos o Comparation Field como Area\_Under\_ROC.

Relatório de Teste Estatístico

O teste estatístico realizado foi o PairedCorrectedTTester do pacote Weka. O objetivo do teste foi comparar o desempenho de diferentes algoritmos em relação à métrica "Area\_under\_ROC" em um único conjunto de dados.

Foram utilizados os seguintes parâmetros no teste:

* Datasets: 1 conjunto de dados.
* Resultados: 6 resultados obtidos para cada algoritmo.
* Nível de confiança: 0.05 (teste bilateral).

A tabela acima apresenta os resultados do teste para cada algoritmo em relação à métrica "Area\_under\_ROC":

No conjunto de dados "NoShow dataset-weka.filters(100)", os algoritmos 3, 4 e 5 apresentaram um desempenho estatisticamente melhor do que o algoritmo 1 em termos da métrica "Area\_under\_ROC". O algoritmo 2 e 6 não apresentou diferença estatisticamente significativa em relação ao algoritmo 1.

O teste foi realizado considerando um nível de significância de 0.05. A coluna "(v/ /)" indica as diferenças estatisticamente significativas em relação ao algoritmo 1, com "v" indicando que o resultado foi melhor. A coluna "(0/1/0)" mostra o número de comparações significativamente melhores/iguais/piores em relação ao algoritmo 1 para cada algoritmo.

(A) Apresente os resultados e comente que algoritmos foram mais bem sucedidos nos problemas analisados, considerando a métrica da área sob a curva ROC.

No nosso experimento, os algoritmos que tiveram vitória estatística foram os que tem o v, ou seja, o 3 (RandomForest), o 4(iBk com kNN=5) e o 5 (NaiveBayes), pois tiveram sucessivamente os resultados 0.67, 0.59 e 0.63.

(b) Comente que algoritmos foram estatisticamente superiores ao baseline (ZeroR).

-Árvore de decisão (RandomForest):Em relação aos outros o RandomForest teve um desempenho como inidicador maior (0.67) comparado ao ZeroR (0.50) A área sob a curva ROC para a RandomForest de decisão foi de 0,67, o que indica um desempenho estatisticamente superior em comparação ao ZeroR. Isso significa que a RandomForest conseguiu fazer melhores previsões em relação à probabilidade de um evento ocorrer ou não.

-Algoritmo Lazy (IBk): A área sob a curva ROC para o algoritmo Lazy IBk foi de 0,59 que indica, O DESVIO PADRÃO indica estatisticamente melhor. O algoritmo Lazy IBk mostrou um desempenho melhor ao classificar os dados em relação ao evento em questão.

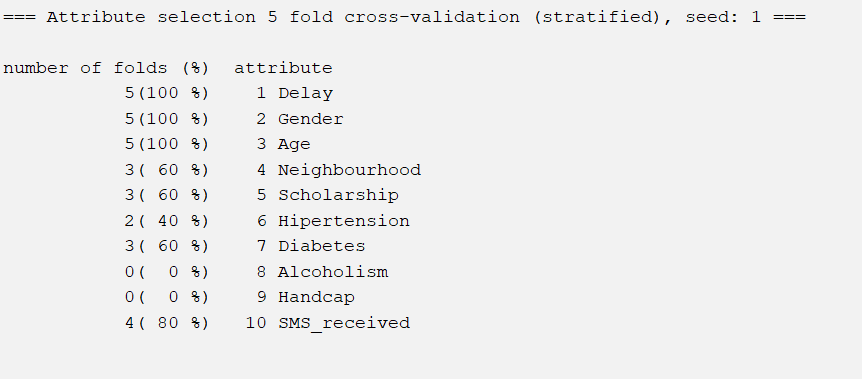
Naive Bayes: A área sob a curva ROC para o Naive Bayes foi de 0,63, indicando um desempenho estatisticamente superior ao ZeroR. O Naive Bayes foi capaz de capturar as relações probabilísticas dos dados e fazer previsões mais precisas em comparação ao ZeroR.

Esses resultados sugerem que, para o conjunto de dados específico analisado, a RandomForest, o algoritmo Lazy IBk e o Naive Bayes são opções mais adequadas em termos de desempenho em relação à métrica "Area\_under\_ROC" em comparação ao algoritmo ZeroR. No entanto, é importante considerar outras métricas e realizar uma análise abrangente antes de selecionar o algoritmo mais adequado para um determinado problema.

Enquanto o SimplesLogistic teve 0.51 porém não é estatisticamente mais bem sucedido por causa da curva gaussiana dele.

E o 2 foi igual.

Questão 4

O algoritmo RandomForest foi o que apresentou os melhores resultados nos experimentos da questão 4 e por isso o escolhemos, para utilizar novamente a técnica de Select Attributes, rodando com as configurações parecidas da questão 2, mudando, apenas, no WrapperSubsetEval: Classifier = RandomForest, temos as seguintes porcentagens para a seleção de atributos: 

Removendo os atributos iguais ou abaixo de 40%, no caso, Hipertension, Alcoholism, Handcap, para tentar aumentar a qualidade dos resultados e obter um limiar de verdadeiros positivos de no mínimo 90% com o maior valor da curva ROC possível, conseguimos o seguinte resultado:

