Nome: Luiza Ávila

Lista 04

1) No artigo “A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data” os autores estudaram o problema de desempenho de sistemas de aprendizagem. Previamente, acreditava-se que o problema estava relacionado ao desequilíbrio de classes, mas os autores chegaram a uma conclusão diferente: parece estar relacionado a aprendizagem com poucos exemplos de classes minoritárias na presença de outros fatores complicadores, como sobreposição de classes. Para chegarem a essa conclusão, realizaram uma ampla avaliação experimental envolvendo dez métodos, três deles propostos pelos autores, em treze conjuntos de dados UCI. Os experimentos comparativos mostraram que, em geral,métodos de oversampling fornecem resultados mais precisosdo que métodos de undersampling, considerando a área soba curva ROC (AUC).

2) No artigo “Estudo Comparativo entre os algoritmos de Mineração de Dados Random Forest e J48 na tomada de Decisão” o objetivo principal colocar os algoritmos de árvore de decisão Random Forest e J48 frente a frente e compará-los. Primeiro, somo introduzidos ao algoritmo Random Forest, sua funcionalidade é explicada e suas principais características são abordadas. Logo após, somos introduzidos ao algoritmo de J48, e somos apresentados com as mesmas informações em relação a ele.Após isso, somos introduzidos a base de dados, declarações de óbitos do estado do Rio Grande do Sul em 2013, e seus atributos. Depois os autores comparam os dois algoritmos usando dados coletados pelo exemplo, por fim, os autores concluírem que o J48 é o melhor algoritmo quando se trata de performance em velocidade ou agilidade na mineração, e o Random Forest é o melhor em relação ao desempenho namineração.

3) Todos os métodos visam minimizar as discrepâncias entre as classes minoritária e majoritária, diminuindo o desbalanceamento.

Redefinir o tamanho de conjunto de dados: oversampling, replicar dados minoritários e undersampling, reduzir número da classe majoritária.

Utilizar diferentes custos de aplicação para diferentes classes: classe minoritária e majoritária possuem “pesos” diferentes.

Induzir um modelo para a classe: as classes são aprendidas separamente, melhorando a aprendizagem.

4)

seed -- Número aleatório de sementes usadas. Default: 1

storeOutOfBagPredictions -- Se deve armazenar as previsões out-of-bag. Default: False

numExecutionSlots -- O número de slots de execução (threads) a serem usados para construir o conjunto. Default: 1

bagSizePercent -- Tamanho de cada bolsa, como uma porcentagem do tamanho do conjunto de treinamento. Default: 100

NumDecimalPlaces O número de casas decimais a serem usadas para a saída de números no modelo. Default:2

batchSize -- O número preferencial de instâncias a serem processadas se a predição em lote estiver sendo executada. Mais ou menos instâncias podem ser fornecidas, mas isso dá às implementações a chance de especificar um tamanho de lote preferido. Default: 100

printClassifiers -- Imprime os classificadores individuais na saída. Default: False

numIterations -- O número de árvores em random forest. Default: 100

debug -- Se definido como verdadeiro, o classificador pode enviar informações adicionais para o console. Default: False

outputOutOfBagComplexityStatistics -- Se deve gerar estatísticas baseadas em complexidade quando a avaliação out-of-bag for realizada. Default: False

breakTiesRandomly -- Rompa os laços aleatoriamente quando vários atributos parecem igualmente bons. Default: False

doNotCheckCapabilities -- Se definido, os recursos do classificador não são verificados antes que o classificador seja construído. Default: False

maxDepth -- A profundidade máxima da árvore, 0 para ilimitado. Default: 0

computeAttributeImportance -- Calcule a importância do atributo por meio da diminuição média da impureza. Default: False

calcOutOfBag -- Se o erro fora da bag é calculado. Default: False

numFeatures -- Define o número de atributos escolhidos aleatoriamente. Se 0, int (log_2 (#predictors) + 1) é usado. Default: 0

5)

A-

ItemSet 1 -

Café 3/10 =0.3

Pão 5/10 = 0.5

Manteiga 5/10 = 0.5

ItemSet 2 -

Café e pão 3/10 = 0.3

Café e manteiga 3/10 = 0.3

Pão e Manteiga 4/10 = 0.4

ItemSet 3 -

Café, pão e manteiga 3/10 = 0.3

Regras
Se Café --> Pão
Se Café --> manteiga
Pão --> manteiga
Manteiga --> Pão
Se Café e pão --> manteiga
Se Café e manteiga --> pão
Se café --> pão e manteiga

C- Weka bateu com o encontrado.

6) Se mudarmos de 0.8 para 0.6 a confiança, nada se altera, pois o suporte mínimo continua o mesmo.

Regras:

Banana=COMPRA 7 ==> Shampoo=COMPRA 7

2. Banana=COMPRA Protetor Solar=NÃO 6 ==> Shampoo=COMPRA 6

3. Protetor Solar=NÃO 8 ==> Shampoo=COMPRA 7

4. Banana=COMPRA 7 ==> Protetor Solar=NÃO 6

5. Escova=COMPRA 7 ==> Batata=COMPRA 6

6. Batata=COMPRA 7 ==> Escova=COMPRA 6

7. Pasta de dente=NÃO 7 ==> Shampoo=COMPRA 6

8. Pasta de dente=NÃO 7 ==> Protetor Solar=NÃO 6

9. Creme para mãos=NÃO 7 ==> Pasta de dente=NÃO 6

10. Pasta de dente=NÃO 7 ==> Creme para mãos=NÃO 6

11. Escova=COMPRA 7 ==> Shampoo=COMPRA 6

12. Creme para mãos=NÃO 7 ==> Shampoo=COMPRA 6

13. Creme para mãos=NÃO 7 ==> Protetor Solar=NÃO 6

14. Shampoo=COMPRA Protetor Solar=NÃO 7 ==> Banana=COMPRA 6

15. Banana=COMPRA Shampoo=COMPRA 7 ==> Protetor Solar=NÃO 6

7)

A) Não está habilitado pois possui atributos numéricos

B) 10 regras foram encontradas. A maioria delas referente ao comprimento da pétala. Também possui em maioria regras que dependem só de um atributo.

C) Mesma coisa que o anterior

```
Best rules found:

 1. petalwidth='(-inf-0.34]' 41 ==> class=Iris-setosa 41    <conf:(1)> lift:(3) lev:(0.18) [27] conv:(27.33)
 2. petallength='(-inf-1.59]' 37 ==> class=Iris-setosa 37    <conf:(1)> lift:(3) lev:(0.16) [24] conv:(24.67)
 3. petallength='(-inf-1.59]' petalwidth='(-inf-0.34]' 33 ==> class=Iris-setosa 33    <conf:(1)> lift:(3) lev:(0.15) [22] conv:(22)
 4. petalwidth='(1.06-1.3]' 21 ==> class=Iris-versicolor 21    <conf:(1)> lift:(3) lev:(0.09) [14] conv:(14)
 5. petallength='(5.13-5.72]' 18 ==> class=Iris-virginica 18    <conf:(1)> lift:(3) lev:(0.08) [12] conv:(12)
 6. sepallength='(4.66-5.02]' petalwidth='(-inf-0.34]' 17 ==> class=Iris-setosa 17    <conf:(1)> lift:(3) lev:(0.08) [11] conv:(11.33)
 7. sepalwidth='(2.96-3.2]' class=Iris-setosa 16 ==> petalwidth='(-inf-0.34]' 16    <conf:(1)> lift:(3.66) lev:(0.08) [11] conv:(11.63)
 8. sepalwidth='(2.96-3.2]' petalwidth='(-inf-0.34]' 16 ==> class=Iris-setosa 16    <conf:(1)> lift:(3) lev:(0.07) [10] conv:(10.67)
 9. petallength='(3.95-4.54]' 26 ==> class=Iris-versicolor 25    <conf:(0.96)> lift:(2.88) lev:(0.11) [16] conv:(8.67)
10. petalwidth='(1.78-2.02]' 23 ==> class=Iris-virginica 22    <conf:(0.96)> lift:(2.87) lev:(0.1) [14] conv:(7.67)
```

8)

A-

```
Best rules found:

 1. biscuits=t frozen foods=t fruit=t total=high 788 ==> bread and cake=t 723    <conf:(0.92)> lift:(1.27) lev:(0.03) [155] conv:(3.35)
 2. baking needs=t biscuits=t fruit=t total=high 760 ==> bread and cake=t 696    <conf:(0.92)> lift:(1.27) lev:(0.03) [149] conv:(3.28)
 3. baking needs=t frozen foods=t fruit=t total=high 770 ==> bread and cake=t 705    <conf:(0.92)> lift:(1.27) lev:(0.03) [150] conv:(3.27)
 4. biscuits=t fruit=t vegetables=t total=high 815 ==> bread and cake=t 746    <conf:(0.92)> lift:(1.27) lev:(0.03) [159] conv:(3.26)
 5. party snack foods=t fruit=t total=high 854 ==> bread and cake=t 779    <conf:(0.91)> lift:(1.27) lev:(0.04) [164] conv:(3.15)
 6. biscuits=t frozen foods=t vegetables=t total=high 797 ==> bread and cake=t 725    <conf:(0.91)> lift:(1.26) lev:(0.03) [151] conv:(3.06)
 7. baking needs=t biscuits=t vegetables=t total=high 772 ==> bread and cake=t 701    <conf:(0.91)> lift:(1.26) lev:(0.03) [145] conv:(3.01)
 8. biscuits=t fruit=t total=high 954 ==> bread and cake=t 866    <conf:(0.91)> lift:(1.26) lev:(0.04) [179] conv:(3)
 9. frozen foods=t fruit=t vegetables=t total=high 834 ==> bread and cake=t 757    <conf:(0.91)> lift:(1.26) lev:(0.03) [156] conv:(3)
10. frozen foods=t fruit=t total=high 969 ==> bread and cake=t 877    <conf:(0.91)> lift:(1.26) lev:(0.04) [179] conv:(2.92)
11. baking needs=t fruit=t vegetables=t total=high 831 ==> bread and cake=t 752    <conf:(0.9)> lift:(1.26) lev:(0.03) [153] conv:(2.91)
12. biscuits=t milk-cream=t total=high 907 ==> bread and cake=t 820    <conf:(0.9)> lift:(1.26) lev:(0.04) [167] conv:(2.89)
13. biscuits=t vegetables=t total=high 950 ==> bread and cake=t 858    <conf:(0.9)> lift:(1.25) lev:(0.04) [174] conv:(2.86)
14. baking needs=t fruit=t total=high 963 ==> bread and cake=t 869    <conf:(0.9)> lift:(1.25) lev:(0.04) [175] conv:(2.84)
15. tissues-paper prd=t fruit=t total=high 878 ==> bread and cake=t 792    <conf:(0.9)> lift:(1.25) lev:(0.03) [160] conv:(2.83)
```

B- Acredito que as regras poderiam ficar melhores se reduzirmos a confiança (no momento eu coloquei elas como 0.9). No momento elas são muito dependentes uma da outra.