Bases ARFF Suportes

João Paulo P. Silva¹, Rosana Lago Kormoczi Lopes¹

¹Análise e Desenvolvimento de Sistemas – Faculdade de Tecnologia de Praia Grande (FATECPG)

Praia Grande – SP – Brasil

joaop\_psilva@outlook.com, rosanalkl@hotmail.com

**Resumo.** Este meta-artigo descreve a classificação feita através do software Eclipse de uma base ARFF nominal chamada ‘Suportes’. Foram utilizados os algoritmos “Naive Bayes” e “Árvore de decisão Id3” com a intenção de obter a melhor classificação da base informada. O Algoritmo que apresentou melhores resultados foi o Naive Bayes.

1. Introdução

A base utilizada, cujos dados são nominais, foi desenvolvida baseado em um problema encontrado entre os jogadores do famoso jogo League of Legends (Riot Games). O jogo conta com mais de 130 personagens (ou campeões como também são chamados), eles possuem histórias diferentes, habilidades únicas e uma determinada classe: mago, atirador, lutador, tanque, assassino ou suporte, sendo essa última a classe onde existem mais possibilidades e mudanças de personagens. Por conta dessa grande variedade na classe “Suporte” jogadores iniciantes (ou até mesmo os experientes) que desejam jogar nesta classe, sentem um pouco de dificuldade na hora de escolher seu campeão, pois eles não tem certeza de qual deles realmente funcionará como suporte (além dos que já são realmente suportes).

Como dito a cima, a classe suporte é a que mais sofre mudanças dentro do jogo, em certo momento bons suportes são aqueles que oferecem curas e escudos aos seus aliados, em outro momento são os personagens que possuem grande defesa e conseguem aguentar uma quantia alta de dano do time inimigo, e em outros bons suportes são aqueles que causam muito dano nos inimigos (como os magos). Tudo depende do momento em que o jogo se encontra ou da formação do seu time. Portanto os atributos escolhidos para classificar quais personagens do jogo podem vir a ser jogados como suportes são:

* Dano {Fraco, Moderado, Forte};
  + Nível de dano do personagem.
* Resistencia {Pouca, Normal, Muita};
  + Se o personagem pode agir como um tanque (muita vida e defesa).
* Controle de Grupo {Ruim, Bom, Ótimo};
  + Se o personagem possui alguma habilidade que prenda ou inutilize o inimigo (Stun, Snare, Blind), ou que diminua sua velocidade de movimento (Slow) etc.
* Mobilidade {Baixa, Media, Alta};
  + Se o personagem possui alguma habilidade que o permita se deslocar rapidamente, desviando de projéteis inimigos (dash, jump).
* Utilidade {Escassa, Razoável, Bastante};
  + Se o personagem possui alguma habilidade que provém curas, escudos, buffs (aumento de dano, de velocidade, de defesa etc) para seus aliados.
* **Classe:** Suporte {Sim, Não};

Esta base será de grande ajuda aos jogadores à medida que novos personagens são inseridos no jogo, pois eles poderão utilizar desta classificação para determinar se estes personagens funcionam como suporte ou não, dessa forma aumentando suas chances de ganharem mais partidas, e jogadores que possuem um número alto de partidas ganhas chamam a atenção do cenário competitivo do jogo podendo até fechar contratos com times profissionais e começar a jogar profissionalmente.

1. Referencial Teórico

## 2.1. Arquivos ARFF

Um arquivo ARFF (Formato de arquivo do atributo-relação) é um arquivo de texto ASCII que descreve uma lista de instancias compartilhando um conjunto de atributos. Arquivos ARFF possuem duas sessões distintas. A primeira sessão é a informação do ‘Cabeçalho’(Header), que é seguida pela informação dos ‘Dados’(Data).

O cabeçalho do arquivo ARFF contém o nome da relação, uma lista dos atributos (as colunas nos Dados), e seus tipos. Um exemplo de Cabeçalho do conjunto de dados padrões ‘IRIS’ é parecido com isso:

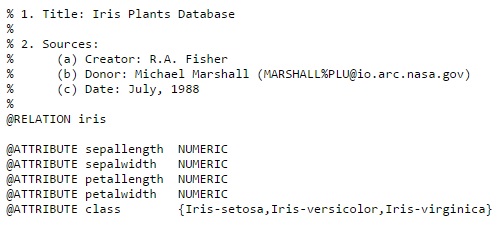


Figura 1. Exemplo de Cabeçalho

Os dados do arquivo ARFF são como os seguintes:

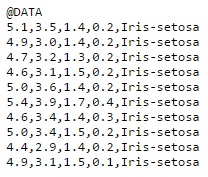


Figura 2. Exemplo de Dados

Linhas que começam com o ‘%’ são linhas comentadas. As declarações @RELATION, @ATTRIBUTE e @DATA são case insensitive.

## 2.2. Atributos Nominais

Para os atributos nominais não existe necessariamente uma ordem dos dados (abecedário não pode ser incluso nessa categoria) são simplesmente nomes diferentes e provêm apenas informação suficiente para distinguir uma instância de outra. Ex.: sexo, cor dos olhos, fumante/não fumante, doente/sadio etc.

## 2.2. Algoritmo “Naive Bayes”

É uma técnica de classificação baseado no teorema de Bayes com uma suposição de independência entre os preditores. Em termos simples, um classificador Naive Bayes assume que a presença de uma característica particular em uma classe não está relacionada com a presença de qualquer outro recurso. Por exemplo, um fruto pode ser considerado como uma maçã se é vermelho, redondo, e tiver cerca de 3 polegadas de diâmetro. Mesmo que esses recursos dependam uns dos outros ou da existência de outras características, todas estas propriedades contribuem de forma independente para a probabilidade de que este fruto é uma maçã e é por isso que é conhecido como ‘Naive’.

*“O classificador é denominado ingênuo (Naive Bayes) por assumir que os atributos são condicionalmente independentes, ou seja, a informação de um evento não ´e informativa sobre nenhum outro. Apesar desta premissa “ingênua” e simplista, o classificador reporta o melhor desempenho em várias tarefas de classificação. (http://www2.dbd.pucrio.br/pergamum/tesesabertas/0420991\_06\_cap\_04.pdf.)”.*

## 2.3. Algoritmo “Árvore de decisão Id3”

As árvores de decisão são representações simples do conhecimento e têm sido aplicadas em sistemas de aprendizado. Elas são amplamente utilizadas em algoritmos de classificação, como um meio eficiente para construir classificadores que predizem classes baseadas nos valores de atributos. Assim, podem ser utilizadas em várias aplicações como diagnósticos médicos, análise de risco em créditos, entre outros exemplos.

*“As árvores de decisão estão entre os mais populares algoritmos de inferência e tem sido aplicado em várias áreas como, por exemplo, diagnóstico médico e risco de crédito, e deles pode-se extrair regras do tipo “se-então” que são facilmente compreendidas (http://www.maxwell.vrac.puc-rio.br/7587/7587\_4.PDF)”.*

1. Resultados Experimentais

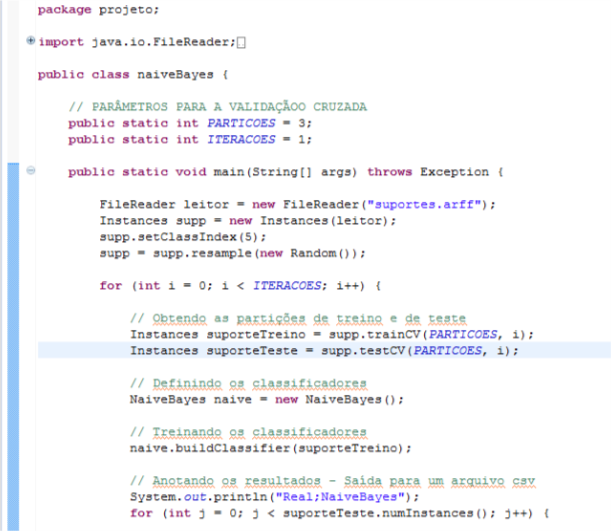
Para a base “Suportes” foi utilizado o algoritmo “Naive Bayes” e o algoritmo “Árvore de decisão (Id3)”, onde foram inseridas 133 instâncias no arquivo ARFF.

Figura 3 - Início da classe naiveBayes

A imagem acima representa a classe responsável pela classificação NaiveBayes, foram utilizadas 3 partições e 1 iteração. No método “setClassIndex” foi utilizado a 6ª linha (5ª posição) dos atributos da base ARFF pois corresponde à classe de nossos atributos.

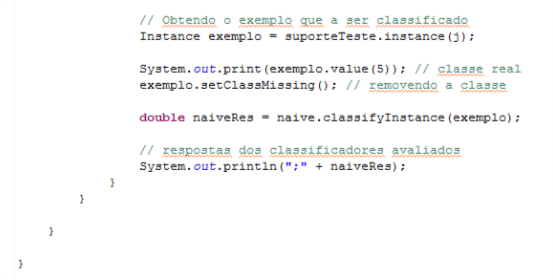
E a imagem abaixo mostra a continuação dessa classe, podemos identificar o uso do método “setClassMissing” cujo objetivo é remover a classe da primeira instância da base que foi escolhida para ser usada como exemplo.

Figura 4 - Continuação da classe naiveBayes

A classe “idtres” que foi utilizada para classificar os dados utilizando o algoritmo “Árvore de decisão” é de extrema semelhança à classe anterior (naiveBayes), única diferença é no nome do classificador e das variáveis, como pode ser visto na imagem abaixo:

Figura 5 – Parte da classe idtres

E para a construção da árvore de decisão, a classe “arvore” foi criada. E da mesma maneira como as classes anteriores, esta classe determina a 5ª posição (6ª linha) como a classe do arquivo com o método “setClassIndex”, além de utilizar o método “firstInstance” para separar a primeira instancia e “setClassMissing” para eliminar a classe dessa instância para poder ser usada como um exemplo, como mostra a imagem a seguir:

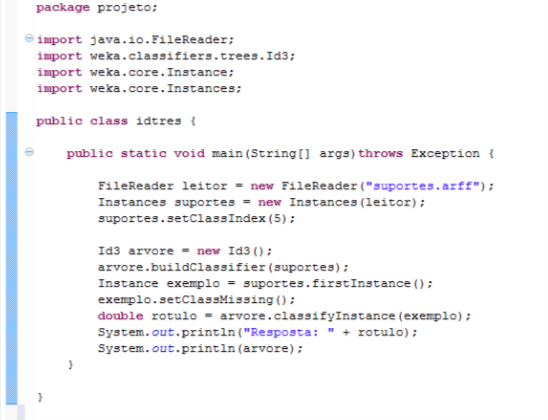


Figura 6 - Classe arvore

E o resultado ao executar esta classe é o seguinte:

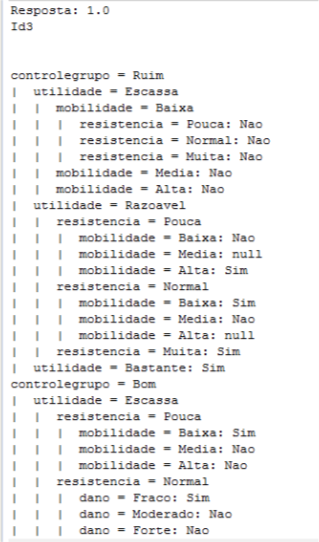
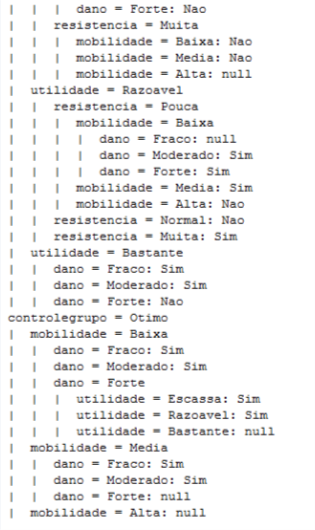


Figura 8 - Árvore de Decisão (2)

Figura 7 - Árvore de Decisão (1)

E ao executar as classes “naiveBayes” e “idtres” os resultados gerados eram copiados para o bloco de notas onde eram salvos como “.csv” parar poderem ser abertos no Excel.

|  |  |
| --- | --- |
| Real | Arvore |
| 1.0 | 1.0 |
| 1.0 | 1.0 |
| 1.0 | 1.0 |
| 0.0 | 1.0 |
| 0.0 | 0.0 |
| 0.0 | 0.0 |
| 0.0 | 0.0 |
| 1.0 | 1.0 |
| 1.0 | 1.0 |
| 1.0 | 1.0 |
| 1.0 | 0.0 |
| 0.0 | 0.0 |
| 1.0 | 1.0 |
| 1.0 | 1.0 |
| 0.0 | 0.0 |
| 1.0 | 0.0 |
| 1.0 | 1.0 |
| 0.0 | 0.0 |
| 1.0 | 0.0 |
| 1.0 | 1.0 |
| 1.0 | 1.0 |
| 0.0 | 1.0 |
| 1.0 | 1.0 |
| 1.0 | 1.0 |
| 1.0 | 1.0 |
| 1.0 | 1.0 |
| 1.0 | 0.0 |
| 0.0 | 0.0 |
| 0.0 | 0.0 |
| 0.0 | 0.0 |
| 1.0 | 1.0 |
| 1.0 | 1.0 |
| 0.0 | 1.0 |
| 0.0 | 0.0 |
| 0.0 | 0.0 |
| 1.0 | 1.0 |
| 0.0 | 1.0 |
| 1.0 | 1.0 |
| 1.0 | 1.0 |
| 0.0 | 0.0 |
| 0.0 | 0.0 |
| 1.0 | 1.0 |
| 0.0 | 0.0 |
| 1.0 | 1.0 |
| 1.0 | 1.0 |

|  |  |
| --- | --- |
| Real | NaiveBayes |
| 1.0 | 1.0 |
| 1.0 | 1.0 |
| 0.0 | 0.0 |
| 1.0 | 1.0 |
| 1.0 | 1.0 |
| 1.0 | 1.0 |
| 0.0 | 0.0 |
| 0.0 | 0.0 |
| 1.0 | 1.0 |
| 0.0 | 0.0 |
| 1.0 | 1.0 |
| 0.0 | 0.0 |
| 1.0 | 1.0 |
| 0.0 | 0.0 |
| 1.0 | 1.0 |
| 1.0 | 0.0 |
| 1.0 | 1.0 |
| 0.0 | 0.0 |
| 0.0 | 0.0 |
| 0.0 | 0.0 |
| 1.0 | 1.0 |
| 0.0 | 0.0 |
| 1.0 | 1.0 |
| 0.0 | 0.0 |
| 1.0 | 1.0 |
| 1.0 | 1.0 |
| 1.0 | 0.0 |
| 1.0 | 1.0 |
| 1.0 | 1.0 |
| 1.0 | 1.0 |
| 1.0 | 1.0 |
| 1.0 | 1.0 |
| 0.0 | 0.0 |
| 1.0 | 1.0 |
| 0.0 | 0.0 |
| 1.0 | 0.0 |
| 0.0 | 0.0 |
| 1.0 | 1.0 |
| 1.0 | 1.0 |
| 0.0 | 0.0 |
| 1.0 | 1.0 |
| 1.0 | 1.0 |
| 0.0 | 0.0 |
| 0.0 | 0.0 |
| 0.0 | 1.0 |

Utilizando e analisando esses dados foi possível criar as matrizes de confusão para ambos os algoritmos, “Árvore de Decisão” e “NaiveBayes”.

|  |  |  |  |
| --- | --- | --- | --- |
| Árvore | Sim | Não | Total |
| Sim | 14 | 4 | 18 |
| Não | 4 | 23 | 27 |
| Total | 18 | 27 | 45 |

|  |  |  |  |
| --- | --- | --- | --- |
| NaiveB | Sim | Não | Total |
| Sim | 17 | 3 | 20 |
| Não | 1 | 24 | 25 |
| Total | 18 | 27 | 45 |

E com os dados das duas tabelas a cima foram feitos os cálculos de Precisão, Revocação, Acurácia, Taxa de Erro para ambas.

* **Árvore de Decisão:**

|  |  |  |
| --- | --- | --- |
|  | **Precisão** | **Revocação** |
| **Sim** | 78% | 78% |
| **Não** | 85% | 85% |
| **Media** | 81% | 81% |
|  | **Acurácia** | **Taxa de Erro** |
|  | 82% | 18% |

* **NaiveBayes:**

|  |  |  |
| --- | --- | --- |
|  | **Precisão** | **Revocação** |
| **Sim** | 94% | 85% |
| **Não** | 89% | 96% |
| **Media** | 92% | 91% |
|  | **Acurácia** | **Taxa de Erro** |
|  | 91% | 9% |

O link para todos esses arquivos, assim como um guia passo-a-passo de como executa-los esta no link a seguir:

<https://github.com/jo-oao/InteligenciaArtificial.git>

1. Conclusão

Conclui-se que o melhor classificador para a base nominal “Suportes” foi o algoritmo “Naive Bayes”, pois ele apresenta resultados maiores que os da “Árvore de Decisão”. Sua Precisão tem uma média de 92%, enquanto a da Árvore é de 81%. Em Revocação Naive Bayes também se encontra na frente com uma média de 91% contra 81% da Árvore, além de que sua acurácia chegou a 91% e a da árvore somente 82%, e sua Taxa de Erro foi de somente 9% enquanto a taxa de erro da árvore de decisão é o dobro, 18%.

Referências

Weka (2002) “Attribute-Relation File Format (ARFF)”, <http://www.cs.waikato.ac.nz/ml/weka/arff.html>, Setembro.

Silvia (2012) “Tipos de variáveis”, <http://leg.ufpr.br/~silvia/CE055/node8.html>, Setembro.

NAIVE BAYES, http://www2.dbd.puc-rio.br/pergamum/tesesabertas/0420991\_06\_cap\_04.pdf.

ÁRVORE DE DECISÃO ID3, http://www.maxwell.vrac.puc-rio.br/7587/7587\_4.PDF.