Laboratório 3 - Árvore de decisão

Disciplina CTC-17 Inteligência Artificial

Prof. Paulo André Lima de Castro

Igor Mourão Ribeiro[1]
Isabelle Ferreira de Oliveira[1]
[1]Aluno de Graduação em Engenharia do Instituto Tecnológico de Aeronáutica (ITA)

E-mail: igormr98mr@gmail.com
isabelle.ferreira3000@gmail.com

Os códigos elaborados podem ser consultados integralmente em **https://github.com/isabelleferreira3000/ctc-17/tree/master/lab3.** Abaixo seguem os resultados obtidos por meio dos códigos implementado. Os arquivos com os códigos também estão em anexo à submissão.

Objetivos

O trabalho tem como objetivo exercitar e fixar conhecimentos adquiridos sobre Árvores de decisão utilizando uma base de dados de fonte diversa e que necessita pré-processamento. Para isso será resolvido o problema proposto: dadas informações sobre um novo usuário (idade, gênero e ocupação), sugerir três filmes que ele irá apreciar. A linguagem escolhida para resolução dos problema foi python.

Descrição dos Classificadores

Assim como foi sugerido no roteiro do laboratório, foi utilizado o banco de dados de classificações de filmes fornecido, assim como foi implementado uma árvore de decisões como modelo de aprendizado.

A forma pensada para realizar o objetivo foi: dado um usuário e um filme (ou seja, as informações de idade, gênero, ocupação, para o usuário; e de gêneros, para o filme), a árvore de decisão chega a um valor de avaliação do filme (um *rating* de 1 a 5). Para recomendar os três filmes, então, percorre-se a lista de filmes, predizendo a avaliação que o usuário em questão o daria, tendo em vista os gêneros do filme e da árvore implementada. Recomenda-se, por fim, os três primeiros filmes com notas 4 ou 5 que aparecem dessa forma.

A implementação da árvore se deu através do algoritmo apresentado em aula e mostrado na Figura 1. Para se chegar aos dados utilizados como parâmetro *exemplos* do algoritmo, foi realizado inicialmente um pré-processamento dos dados fornecidos:

- Primeiro, mergeou-se todas as três tabelas (movies, ratings e users) pelos atributos UserID e MovieID. Retirou-se também as colunas Timestamp e Zip-code por entender-se que são dados irrelevantes para a predição. A essa tabela foi dado o nome de *mergerd_data*.

- A partir de *merged_data*, separou-se gêneros compostos de filmes (por exemplo, “Animation|Children's|Comedy”) em linhas diferentes, cada linha para um gênero. Assim, um usuário avaliando como 5 um filme de “Animation|Children's”, conta como uma avaliação 5 para um filme de “Animation” e uma outra avaliação 5 para um filme de “Children's”. Isso foi feito para tornar mais significativa o atributo *Genrer*, pois inicialmente pouco se repetiam os gêneros compostos. A essa tabela foi dado o nome de *data*.
- Por fim, embaralhou-se as linhas de *data* aleatoriamente, a fim de não prejudicar o treino e o teste quando forem separados os sets de treino e de teste.

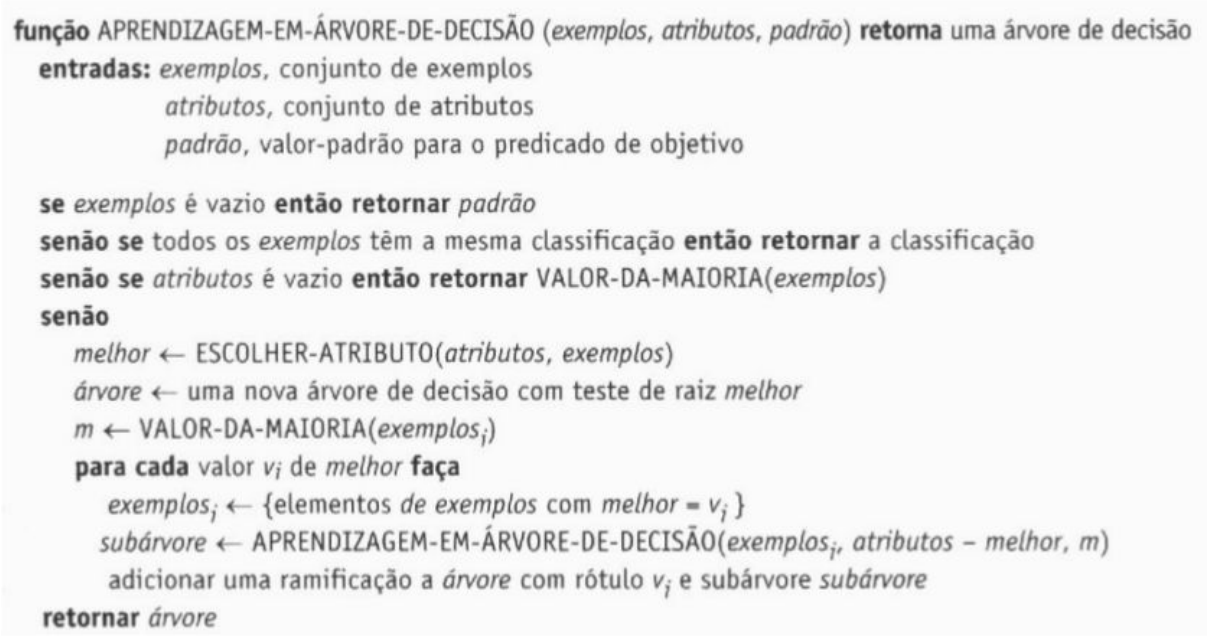

Figura 1: Algoritmo para implementação da árvore de decisão.

Separou-se o dataset final *data* em três subgrupos, *training set*, *cross validation* set e *test set*, seguindo uma proporção de 3:1:1. Numa primeira situação, utilizou-se todo a tabela *data* (cerca de 1.200.000 linhas para o *training set*); já numa segunda situação, utilizou-se apenas cerca de 50.000 linhas para o *training set*.

No classificador *a priori*, apenas avaliou-se os filmes seguindo sempre com a avaliação mais recorrente.

Dados e Resultados da comparação

Os dados foram melhor descritos na seção Descrição dos Classificadores acima, mas a seguir também será abordado o *dataset* criado pelas avaliações de filme da aluna Isabelle. Já os resultados para os modelos descritos anteriormente foram apresentados nas Tabela de 2 a 5.

Acerca do dataset criado pelas avaliações de filme da aluna Isabelle, foram analisados 10 filmes. A lista de filmes com a avaliação da aluna e a predição da árvore de decisão foi apresentada na Tabela 1. As cores na Tabela 1 foram escolhidos da seguinte

maneira: em verde, o caso de acerto; em vermelho, predições completamente erradas (na qual um filme seria indicado e o usuário não gostaria, ou um filme que o usuário gostaria bastante não foi indicado); em amarelo, predições erradas, porém não tão prejudiciais (uma vez que o filme seria indicado e o usuário gostaria do filme, mesmo que não no mesmo nível predito).

Tabela 1: Filmes, avaliações e predições para o dataset criado pela aluna.

Filme	Avaliação	Predição
1: Toy Story (1995)	5	3
3799: Pok◆mon the Movie 2000 (2000)	5	3
3945: Digimon: The Movie (2000)	4	3
3752: Me, Myself and Irene (2000)	3	4
3564: Flintstones in Viva Rock Vegas, The (2000)	3	3
3527: Predator (1987)	2	4
2205: Mr. & Mrs. Smith (1941)	2	4
2959: Fight Club (1999)	5	4
19: Ace Ventura: When Nature Calls (1995)	3	4
2571: Matrix, The (1999)	5	4

Tabela 2: Acurácias dos modelos apresentados.

Modelo	Acurácia		
	Training set	**Validation set**	**Test set**
***Decision Tree* com todos os dados**	0.3552	0.3498	0.3493
***Decision Tree* com parte dos dados**	0.3999	0.3235	0.3209
***A priori* com avaliação mais recorrente**	0.3476	0.3446	0.3449
Decision Tree com	-	-	0.15

dataset criado pela aluna Isabelle			
***A priori* com dataset criado pela aluna Isabelle**	-	-	0.15

Tabela 3: Matriz de confusão dos modelos apresentados.

Modelo	Matriz de confusão		
	Training set	**Validation set**	**Test set**
***Decision Tree* com todos os dados**	[[1088 85 8141 58216 2334] [410 308 14395 117128 4128] [649 186 32479 286089 11303] [581 177 26947 391793 17870] [419 101 16473 247040 22749]]	[[269 20 2756 19503 817] [160 74 4783 39097 1396] [226 84 10166 96134 3844] [241 72 9842 129201 6409] [158 36 5611 82461 7003]]	[[352 32 2697 19493 800] [133 78 4750 39066 1318] [236 105 9956 95752 3780] [190 83 9815 129328 6516] [176 37 5713 82911 7046]]
***Decision Tree* com parte dos dados**	[[136 38 549 1817 317] [30 241 1009 3451 592] [30 53 3586 8287 1350] [55 80 2178 13390 1833] [27 67 1522 6829 2976]]	[[20 12 198 583 106] [9 25 383 1221 214] [28 47 845 2990 600] [23 54 968 3914 835] [14 38 622 2429 636]]	[[11 13 202 589 101] [12 24 377 1290 214] [31 38 829 2868 572] [32 54 960 3906 848] [14 37 605 2538 649]]
***A priori* com avaliação mais recorrente**	[[0 0 0 2857 0] [0 0 0 5323 0] [0 0 0 13306 0] [0 0 0 17536 0] [0 0 0 11421 0]]	[[0 0 0 919 0] [0 0 0 1852 0] [0 0 0 4510 0] [0 0 0 5794 0] [0 0 0 3739 0]]	[[0 0 0 916 0] [0 0 0 1917 0] [0 0 0 4338 0] [0 0 0 5800 0] [0 0 0 3843 0]]
Decision Tree com dataset criado pela aluna Isabelle	-	-	[[0 2 2 0] [1 0 3 0] [1 2 0 0] [2 4 3 0]]
***A priori* com dataset criado pela aluna Isabelle**	-	-	[[0 0 4 0] [0 0 4 0] [0 0 3 0] [0 0 9 0]]

Tabela 4: Erro quadrático médio dos modelos apresentados.

Modelo	**Erro quadrático médio**		
	Training set	**Validation set**	**Test set**
***Decision Tree* com todos os dados**	1.4430	1.4571	1.4541
***Decision Tree* com parte dos dados**	1.5159	1.6776	1.6866
***A priori* com avaliação mais recorrente**	1.4220	1.4231	1.4329
Decision Tree com dataset criado pela aluna Isabelle	-	-	2.0
***A priori* com dataset criado pela aluna Isabelle**	-	-	1.45

Tabela 5: Estatística kappa dos modelos apresentados.

Modelo	**Estatística kappa**		
	Training set	**Validation set**	**Test set**
***Decision Tree* com todos os dados**	0.03269	0.02301	0.02275
***Decision Tree* com parte dos dados**	0.1322	0.0204	0.01914
***A priori* com avaliação mais recorrente**	0.0	0.0	0.0
Decision Tree com dataset criado pela aluna Isabelle	-	-	-0.07594
***A priori* com dataset criado pela aluna Isabelle**	-	-	0.0

Discussão e sugestão de melhorias para o classificador

Acreditou-se que os resultados para acurácia (assim como para as outras métricas) não foram tão satisfatórios devido principalmente a presença de ruídos nos dados. Esses ruídos são ocasionados por diferentes usuários acabarem se encaixando no mesmo perfil

(Idade, Gênero e Ocupação) e avaliarem filmes diferentemente. Além disso, separar os gêneros de filmes compostos em linhas distintas com gêneros simples também contribuiu para o aumento do ruído. Utilizar os nomes dos filmes infelizmente era inviável, dada a grande quantidade de filmes diferentes.

Através da matriz de confusão, pode-se perceber a grande tendência em avaliar-se com notas altas (4 e 5) em detrimento das demais notas. O erro quadrático médio (assim como a acurácia) mostrou que o modelo muito mais simples (o *a priori*) alcançou resultados semelhantes, ou até mesmo melhores do que os da árvore de decisão. Infelizmente, o coeficiente Kappa apresentou o quão semelhante ao aleatório acabou se tornando essa previsão.

Esse resultado, entretanto mede referente aos acertos das previsões de avaliações dadas aos filmes pelos usuários, o que não necessariamente prejudica a indicação de filme tanto assim. Note que, caso o sistema preveja avaliações 4, 4 e 5 para supostos três filmes, e o usuário passa a avaliar como 5, 5, 4, não houve nenhum acerto usando essas métricas, mas mesmo assim as indicações deixaram o usuário satisfeito.

Seria interessante, então, analisar mais detalhadamente o resultado do *test set*, assim como foi analisado com as cores a Tabela 1. Assim, talvez possa-se observar que as previsões de avaliações, embora erradas, não prejudicasse tanto o usuário. Por exemplo, para o caso da Tabela 1, seriam indicados para se assistir os filmes “Me, Myself and Irene (2000)”, “Predator (1987)”, “Mr. & Mrs. Smith (1941)”, “Fight Club (1999)”, “Ace Ventura: When Nature Calls (1995)” ou “Matrix, The (1999)”, e a usuária estaria satisfeita com 4 dos 6 filmes indicados.

Conclusões

Proposta de trabalho foi interessante por levar os alunos a implementar um aprendizado por árvore de decisão (o que leva a um entendimento melhor do algoritmo), além de fazê-lo lidar com a preparação inicial dos dados (etapa bastante presente em projetos reais de aprendizado). A dupla classifica esse trabalho como extenso, mas está com a sua complexidade dentro do esperado para o problema.

O trabalho ajudou no entendimento sobre o assunto, servindo não só para mostrar como é a implementação desses algoritmos, mas também entender que às vezes modelos mais simples podem ser semelhantes à modelos mais elaborados. Foi bastante interessante criar o próprio *dataset* com avaliações de filmes, a fim de se colocar no lugar do usuário do serviço fornecido e analisá-lo mais pessoalmente do que só através das métricas numéricas.

Descrição da Implementação

A linguagem escolhida para resolução dos problema foi Python3, utilizando também as bibliotecas Pandas e Sklearn para lidar com arquivos de dados e para as métricas de comparação, respectivamente. A árvore foi implementada sem o auxílio de bibliotecas e pode ter seu código analisado tanto por meio do código enviado em anexo, como pelo link para o repositório Github apresentado no início do relatório.