Algoritmos de Classificação Aplicados em Dados de Filmes

Daniel Lucio Masselani de Moura
Departamento de Computação
Universidade Federal de São Carlos
São Carlos, Brasil
743525

Isaac Willian Mariotto Marques
Departamento de Computação
Universidade Federal de São Carlos
São Carlos, Brasil
754755

Fernando Sassi Nunes
Departamento de Computação
Universidade Federal de São Carlos
São Carlos, Brasil
754754

Felipe Huvos Ribas
Departamento de Computação
Universidade Federal de São Carlos
São Carlos, Brasil
386138

Julio Cesar dos Santos Oliveira Filho
Departamento de Computação
Universidade Federal de São Carlos
São Carlos, Brasil
779800

***Resumo*—O presente estudo apresenta a aplicação dos métodos de Àrvore de Decisão e Vizinhos Mais Próximos para predizer notas de filmes em um intervalo de ”1” a ”5”. O dataset utilizado possui diversos parametros, com valores numéricos e em texto, simples e compostos, por isso foi necessário a realização de um tratamento para adequar os dados.Em termos de resultados, a acurácia obtida por meio da Árvore de Decisão foi a mais alta, na ordem de 60%, especialmente depois de otimizados os hiperparâmetros do modelo. No entanto, o Vizinhos Mais Próximos apresenta acurácia próxima a 50%, o que também não é desprezível dada a quantidade de categorias de nota possíveis - um total de cinco.**

***Termos de indíce*—machine learning, artificial intelligence, decision tree, nearest neighbors, classification, movies**

I. Introdução

Na área de Aprendizado de Máquina, existem diversos problemas de classificação conhecidos e estudados. Um deles é a classificação de dados de modo supervisionado, onde são utilizados algoritmos de classificação chamados também de estimadores. Este trabalho envolve a classificação de filmes de um dataset público em cinco categorias de notas. As categorias de notas utilizadas na classificação dos filmes foram criadas a partir de avaliações dadas aos filmes por pessoas que assistiram aos mesmos, indo de um intervalo de ”1” até ”5”.

Para realizar a classificação dos filmes, foram utilizados dois estimadores distintos da biblioteca python scikit-learn [2]: DecisionTreeClassifier e KNeighborsClassifier. O primeiro estimador implementa a abordagem de árvores de decisão e o segundo se refere à técnica de vizinhos mais próximos (KNN). Usando estes classificadores é possível estimar a categoria de notas em que um determinado filme se encaixa.

A próxima seção deste trabalho contém uma síntese do problema abordado, seguida de uma seção onde os dados utilizados serão descritos. Na seção IV é feita a descrição do uso dos estimadores já apresentados na classificação dos dados sobre filmes. Para finalizar o trabalho, há uma seção onde discute-se os resultados dos estimadores e o impacto de hiperparâmetros associados aos estimadores, além da apresentação das matrizes de confusão obtidas. Por fim, na ultima seção, conclui-se o resultado do trabalho.

II. Apresentação do Problema

Em Aprendizado de Máquina existe a classe dos problemas de classificação supervisionada, onde um estimador é treinado visando a classificar um conjunto de exemplos em categorias pré definidas. No trabalho apresentado o problema abordado é a classificação de filmes em cinco categorias de notas no intervalo de ”1” a ”5”. Para essa classificação ocorrer foram utilizados diversos colunas - features - com a finalidade de fornecer insumos para o modelo tomar a decisão. Vale observar que o dataset de filmes utilizados não possuía os dados prontos para o uso desejado, foi necessária a remoção de algumas colunas não adequadas e tratamento de outras para serem utilizadas pelos estimadores.

III. Descrição dos Dados

O conjunto de dados utlizado contém metadados de 45 mil filmes lançados até Julho de 2017. O dataset possui as seguintes colunas:

- id - identificador do filme no dataset
- imdb_id - identificador do filme no Internet Movie Database (IMDB)
- budget - orçamento do filme
- genres - gêneros associados ao filme (comédia, terror, romance, etc)
- homepage - link para a página do filme
- adult - indicação de conteúdo adulto
- belongs_to_collection - coleção a qual o filme pertence (sagas, trilogia, etc)
- original_language - idioma original
- original_title - título original

- overview - sinopse do filme
- popularity - popularidade do filme
- poster_path - caminho para o poster do filme no The Movie Database (TMDB)
- production_companies - companias que produziram o filme
- production_countries - países onde o filme foi produzido
- release_date - data de lançamento
- revenue - receita gerada pelo filme
- runtime - duração do filme em minutos
- spoken_languages - idiomas falados no filme
- status - status do filme ('Lançado', 'Rumor', 'Pós-Produção', 'Em produção', 'Planejado' ou 'Cancelado')
- tagline - tagline (slogan) do filme
- title - título do filme no IMDB
- video - descrição não encontrada
- vote_average - média das avaliações do filme
- vote_count - número de avaliações

IV. Métodos Utilizados

A. Pré-processamento dos dados

Foi utilizado a biblioteca pandas [1] para carregar o dataset e muitos métodos utlizados no pré-processamento dos dados são dessa biblioteca.

Como explicado na seção anterior, algumas colunas de dados foram removidas, pois continham dados não adequados para o estimador. Dentre essas colunas, temos: "adult", "belongs_to_collection", "homepage", "id", "imdb_id", "original_title", "overview", "poster_path", "tagline", "title", "video", "spoken_languages", "release_date". Também foi removida uma linha em que as posições dos valores não correspondiam às colunas corretas.

As colunas que permaneceram, porém com tratamento, foram: "release_date", originalmente no formato YYYY-MM-DD (ano, mês e dia), onde foram criadas duas novas, chamadas "release_year" para o ano e "release_month" para o mês, "genres", onde foi utilizado apenas o id do primeiro gênero apresentado. Nas colunas "production_companies", "production_countries", "original_language" e "status", foi utilizado somente o primeiro valor da coluna e utilizado um método para transformar os valores de texto para valores numéricos - pandas.factorize-. Já as colunas: "budget" , "popularity", "runtime", "revenue" e "vote_count" não precisaram de tratamento pois já possuíam seus valores como numéricos.

Também foi executada a função *fillna*, do prõprio pandas, para tratamento de valores nulos - tais valores foram alterados de "NaN" para -1.

Durante o tratamento dos dados, foi detectada a situação em que um filme com nenhuma nota recebida estava com a média de notas com valores entre 0.0 e 0.5. Embora tal situação não seja adequada, uma intepretação possível é a de que o filme não foi relevante o suficiente para atrair a atenção de um público que estivesse disposto a votar nele. Nesse sentido, foi mantida esta convenção aparentemente adotada pelo *dataset* original. Outro ponto que merece atenção é que a distribuição de classificações obtida com os dados não ficou concentrada em filmes com notas 0.0 e 0.5 - percentes à classe 1. Ou seja, a convenção mencionada anteriormente não parece ter prejudicado a classificação de dados de outras categorias de nota.

Para treinar o modelo, a coluna "vote_average" foi separada do dataset para o treinamento e os seus valores foram separadas em 5 classes seguindo o seguinte critério:

$$Y = \begin{cases} 1, & \text{if } vote\_average < 2 \\ 2, & \text{else if } vote\_average < 4 \\ 3, & \text{else if } vote\_average < 6 \\ 4, & \text{else if } vote\_average < 8 \\ 5, & \text{otherwise} \end{cases} \quad (1)$$

E então, os dados foram divididos em treino e teste usando a função "train_test_split" do *scikit-learn*.

B. Classificação por Árvore de Decisão

Um método bastante conhecido para classificação de dados de maneira supervisionada é o de árvores de decisão - *Decision Trees*. Este foi o primeiro método utilizado neste trabalho e utiliza a estratégia de divisão e conquista para resolver um problema de decisão. Para classificar um certo exemplo são tomadas diversas decisões, com base nos valores dos parametros dos exemplos, até se obter o rótulo do exemplo. Foram realizados dois tipos de execução deste método: um em que os dados não foram tratados em relação ao balanceamento de exemplos; e uma segunda execução, os dados foram estratificados na divisão do treino e teste, como uma tentativa de ter uma balanceamento de exemplos para cada classe

Também foi usado o método *Grid Search* para estimar os melhores hiperparâmetros para o algorítmo. Sem a estratificação dos dados, os hiperparâmetros escolhidas foram:

- criterion: 'entropy'
- max_depth: 10
- min_samples_leaf: 2
- min_impurity_decrease: 0

Já com estratificação dos dados, os hiperparâmetros selecionados pelo *Grid Search* foram:

- criterion: 'gini'
- max_depth: 10
- min_samples_leaf: 2
- min_impurity_decrease: 0

C. Classificação por Vizinhos Mais Próximos

Outro algorítmo utilizado foi a classificação por Vizinhos mais Próximos - *Nearest Neighbors*, também conhecido pela sigla KNN. A idéia principal deste algoritmo é classificar um novo exemplo com base na sua vizinhança, para isso não é construido um modelo de forma explícita, porém quando um novo exemplo é apresentado ele é classificado com base nos rótulos dos *k* exemplos mais próximos. Primeiramente foi aplicado o algorítmo sem a normalização dos dados. As próximas execuções foram realizadas com os dados normalizados para a mesma escala, para o algorítmo não ficar enviezado pela grandeza dos números de alguma feature.

O *Grid Search* para estimar os melhores hiperparâmetros para o algorítmo.

- metric: 'manhattan'
- n_neighbors: 13
- weights: 'distance'

V. Apresentação dos Resultados

A. Classificação por Árvore de Decisão

A Tabela I mostra as métricas obitidas no caso em que os dados não foram tratados para que ficassem balanceados e onde os hiperparâmetros usados foram os *default* da biblioteca scikit-learn.

Tabela I: Métricas da Árvore de Decisão com parâmetros padrões

Métrica	Valor
Acurácia	53,80%
Precisão	44,13%
Recall	44,65%

Já no caso em que os dados não foram tratados para balanceamento, mas o método *Grid Search* foi utilizado para obtenção de hiperparâmetros otimizados, foram obtidas as métricas da Tabela II.

Tabela II: Métricas da Árvore de Decisão com parâmetros determinados no Grid Search

Métrica	Valor
Acurácia	62,12%
Precisão	54,65%
Recall	46,06%

A Tabela III contém as métricas para a situação em que os dados foram tratados para ficarem balanceados com o uso do método *stratify* e em que o *Grid Search* foi utilizado em conjunto no tratamento.

A Figura 1 mostra as matrizes de confusão para a execução com os parâmetros determinados pelo Grid Search, sem a estratificação dos dados, modelo que recebeu o melhor resultado. A primeira matriz é referente a matriz de confusão com os números absolutos e a segunda, com os números normalizados de 0 a 1.

B. Classificação por Vizinhos Mais Próximos

Considerando a abordagem de classificação por vizinhos mais próximos, foram obtidos diferentes resultados de acordo com a normalização dos dados e/ou a variação do valor do hiperparâmetro *k*.

Sem realizar normalização das features e com valor de *k*=3, a acurácia observada foi de, aproximadamente, 46,74%. A Tabela IV contém as métricas para esse caso.

Já com *k*=15 e, novamente, sem normalização, a acurácia observada foi de, aproximadamente, 51,20%. A Tabela V contém as métricas para esse caso.

Tabela III: Métricas da Árvore de Decisão com parâmetros determinados no Grid Search e dados estratificados

Métrica	Valor
Acurácia	60,66%
Precisão	50,16%
Recall	44,00%

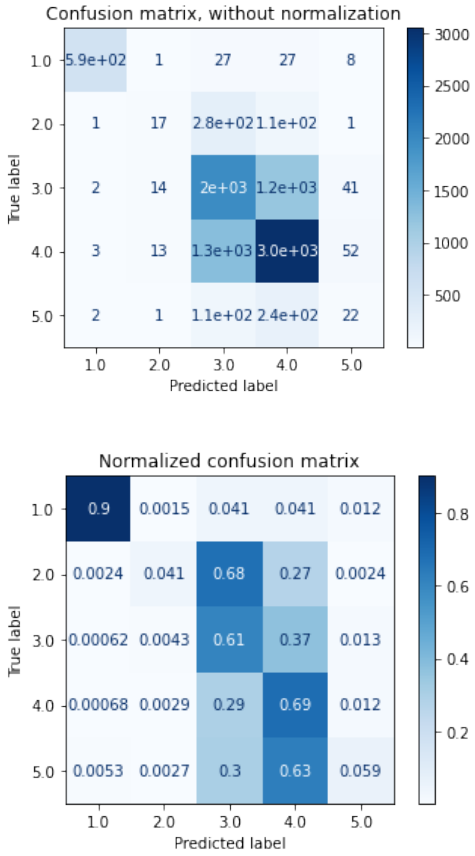

Fig. 1: Matriz de confusão para a árvore de decisão com os hiperparâmetros escolhidos pelo Grid Search

Aplicando normalização nas features e com k=3, a acurácia obtida foi de 46,12%, aproximadamente. A Tabela VI contém as métricas para esse caso.

Por fim, aplicando normalização nas features e com k=15, a acurácia obtida foi de 52,16%, aproximadamente. A Tabela VII contém as métricas para esse caso.

A última execução que realizamos para a abordagem de vizinhos mais próximos contemplou a normalização, porém,

Tabela IV: Vizinhos mais próximos - k=3

Métrica	Valor
Acurácia	46,74%
Precisão	30,68%
Recall	31,75%

Tabela V: Vizinhos mais próximos - k=15

Métrica	Valor
Acurácia	51,20%
Precisão	36,00%
Recall	29,18%

Tabela VI: Vizinhos mais próximos, com dados normalizados - k=3

Métrica	Valor
Acurácia	46,12%
Precisão	30,25%
Recall	30,46%

diferentemente das execuções anteriores, não foi especificado manualmente um valor do hiperparâmetro *k*. O valor de *k* foi obtido por meio do método *Grid Search*, resultando em k=13, já mencionado anteriormente. Neste caso, a acurácia observada foi a mais alta entre as obtidas com a abordagem de vizinhos mais próximos, aproximadamente 52,58%. A Tabela VIII contém as métricas para esse caso.

A Figura 2 mostra as matrizes de confusão para a execução com os parâmetros determinados pelo Grid Search, com normalização dos dados. A primeira matriz é referente a matriz de confusão com os números absolutos e a segunda, com os números normalizados de 0 a 1.

VI. Discussão

A. Classificação por Árvore de Decisão

Na classificação realizada com o uso de Árvore de Decisão, foi obtido um valor de acurácia relativamente alto - aproximadamente 53,80% - logo na primeira execução, na qual não foram realizados tratamentos como balanceamento de *features* e otimização de hiperparâmetros por meio de *Grid Search*.

Utilizando o método de *Grid Search* sem o balanceamento de *features*, foi observada a maior acurácia deste trabalho, de aproximadamente 62,12%. Tal acurácia é especialmente relevante quando considerado que as classes de saída totalizam cinco opções. Ou seja, uma alocação aleatória de valores de saída resultaria em um acerto de 20%, valor muito inferior a este observado.

Tabela VII: Vizinhos mais próximos, com dados normalizados - k=15

Métrica	Valor
Acurácia	52,15%
Precisão	34,80%
Recall	27,40%

Tabela VIII: Vizinhos mais próximos, com dados normalizados e grid search - k=13

Métrica	Valor
Acurácia	52,58%
Precisão	37,65%
Recall	29,77%

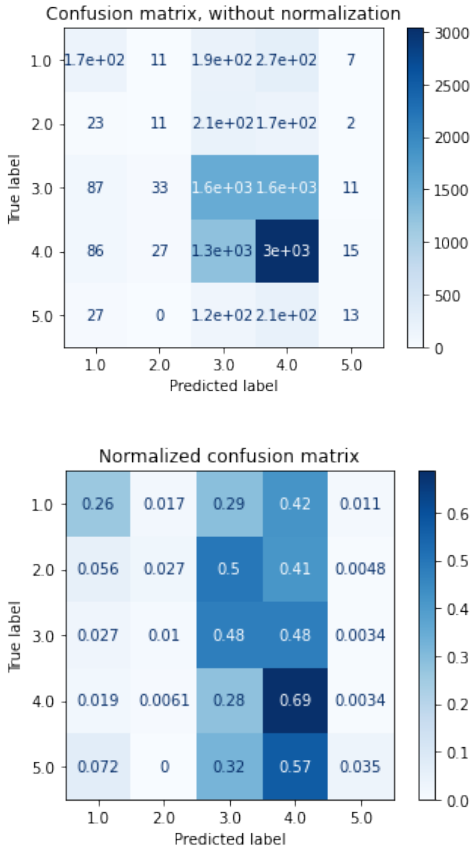

Fig. 2: Matriz de confusão para vizinhos mais próximos com dados normalizados e com os hiperparâmetros escolhidos pelo Grid Search

Curiosamente, a aplicação de balanceamento de *features* associada ao *Grid Search* resultou em uma acurácia inferior - aproximadamente 60,66% - à mencionada anteriormente, embora tenha sido a segunda maior neste trabalho.

B. Classificação por Vizinhos Mais Próximos

A abordagem de classificação por vizinhos mais próximos apresentou os melhores resultados de acurácia na execução em que mais otimizações e/ou tratamentos de dados foram realizados, ou seja, uma situação um pouco mais próxima da esperada. Esta execução envolveu a normalização das *features* e o uso de *Grid Search*, o qual sugeriu o valor 13 para o hiperâmetro *k*.

A execução imediatamente anterior à mencionada acima, que não contou com otimização via *Grid Search*, não resultou em uma acurácia extremamente inferior ao melhor resultado obtido com vizinhos mais próximos. Na verdade, a acurácia obtida ficou bem próxima - 52,15%.

De acordo com os resultados, o maior impacto no valor da acurácia parece ocorrer quando o valor do hiperparâmetro *k* é ajustado, ao menos quando comparados os cenários em que apenas *k* foi alterado de 3 para 15, mantidos os demais

parâmetros ou tratamentos constantes. Por exemplo, sem realizar normalização de *features* e com k no valor 3, a acurácia foi de 46,74%.

C. Classificação por Vizinhos Mais Próximos

Foi observado, nas imagens obtidas de matrizes de confusão, que há uma grande concentração de acertos nas categorias 3 e 4 de notas de filmes. Uma possibilidade para esta situação seria valores mais adequados para a predição de nota no caso de filmes associados a essas categorias.

VII. Conclusão

Árvores de decisão se mostraram uma técnica bastante interessante para a predição de notas - ou classes de notas - de filmes, com base em características diversas desses filmes.

Além disso, o pré-tratamento de dados foi um passo essencial para que os algoritmos de classificação baseados em aprendizado de máquina pudessem ser usados. A decisão de selecionar apenas o primeiro valor disponível em algumas colunas, por exemplo, pareceu acertada: seria difícil, ou até mesmo injusto, comparar filmes com quantidades diferentes de gêneros ou de empresas produtoras. Porém, esta característica dos dados pode ser melhor explorada em um trabalho futuro.

Embora as árvores de decisão tenham mostrado resultados melhores em geral, o desempenho obtido com a técnica KNN não pode ser desprezado - ele foi consideravelmente superior ao que seria obtido por meio de uma classificação aleatória dos filmes em classes de notas. No entanto, cabe observar que a classificação 1 na abordagem KNN teve uma performance bastante inferior. Levando-se em conta apenas as métricas de precisão e *recall*, o KNN teve uma performance significativamente inferior também quando comparado às árvores de decisão.

Por fim, os resultados observados foram bastante interessantes dadas as condições dos dados de origem e o número de observações disponiveis, além da forma com tais observações estavam distribuídas.

Referências

[1] The pandas development team. pandas-dev/pandas: Pandas, February 2020.

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.