Relatório Projeto 3 de Ciência de Dados
Rating de filmes (IMDB)

por: Marcelo Lisboa, Tiago Bergamo e Bruno Kaczelnik - 2C Engenharia Insper

Esse projeto se trata de um classificador, o qual é baseado no elenco, diretores, gêneros do filme, bilheteria e entre outras variáveis, pretendemos avaliar qual será a classificação que um filme receberá.

Para isso, foi selecionado o DataSet “imdb-5000-movie-staset”, encontrado no site Kaggle, no qual se encontram aproximadamente 3760 filmes, cada um possuindo três atores principais, um diretor, diversos gêneros, entre outros indicadores.

A ideia original do projeto era ser capaz de devolver a nota exata que um filme receberia utilizando regressão, entretanto, devido ao curto intervalo no qual a nota poderia ser dada, zero a dez, o modelo seria incapaz de acertar com precisão, ocorrendo diversas extrapolações. Assim, resolvemos utilizar classificação e para isso dividimos os filmes dentre 5 divisões (Muito Bom, Bom, Neutro, Ruim e Muito Ruim).

Foi necessário então fazer uma análise exploratória dos dados desse dataset para selecionar apenas as variáveis que seriam úteis para o nosso modelo de Rating de filmes. A análise segue abaixo:

Análise exploratória:

Inicialmente haviam 5043 filmes com 27 colunas cada, e apenas avaliando o nome de algumas dessas colunas já foi possível perceber se seriam irrelevantes ou não para o nosso modelo. Assim já foram descartados os indicadores 'aspect_ratio', 'movie_imdb_link', 'plot_keywords' e 'facenumber_in_poster'.

Em seguida, para analisar melhor o restante dos dados foram plotados pairplots (gráficos que plotam relações entre os dados do dataset) e uma tabela de correlações. Assim, esses pairplots, vimos que diversos indicadores têm em grande maioria, principalmente os relacionados

com likes, valores iguais a zero, o que possibilitou descartá-los por insuficiência de dados para análise. Além disso, agora olhando as correlações, podemos perceber que tanto o "actor_1_facebook_likes" e "cast_total_facebook_likes" como o "num_voted_users" e "num_user_for_reviews" estão com correlações altíssimas, o que quer dizer que podemos desconsiderar uma das colunas, pois uma pode explicar a outra.

Utilizando agora o value_counts do pandas para ver a proporção dos indicadores, novamente por insuficiência de dados, resolvemos que seria melhor descartar as colunas "color", "language" e "country".

- *Sobre os indicadores selecionados:*

Foi possível observar pelos gráficos plotados que dentre o indicador "num_critic_for_review" e o "num_user_for_review", percebemos que quanto maior o número de críticas e usuários por review, melhor tende a ser a nota do filme. Isso acontece pois, na maioria dos casos, filmes bons acabam recebendo mais atenção dos espectadores, que acabam comentando várias vezes sobre o mesmo sobre os comentários dos outros.

Analisando também a duração de cada filme, vimos que ela influencia em sua nota. É visível que os filmes com uma duração média de 150 minutos são os que possuem notas mais altas. Porém, pode ser observado que há filmes com uma maior e menor duração que também receberam notas elevadas.

Já sobre a receita bruta ("gross") de cada filme, vimos que filmes com receitas brutas maiores tendem a receber notas maiores do que filmes com receitas menores, algo relativamente intuitivo de se pensar.

Agora sobre o ano de lançamento dos filmes, pode-se observar que há uma maior quantidade de dados sobre os filmes mais recentes (1990 em diante), os quais têm uma grande variedade entre notas baixas e altas. Esse cenário é bem diferente do dos filmes mais antigos, os quais se apresentam em menor quantidade no dataset, e os poucos que há, possuem notas, em média, altas, além de ter um maior número de críticas por reviews. Uma possível hipótese que possa explicar esse cenário é que antigamente, quando a indústria de cinema era menor, havia menos filmes sendo lançados anualmente, diferentemente de hoje em dia. Além disso, outra coisa que diferencia os dias atuais de antigamente, é a produção de filmes que recebem notas baixas, como de terror, "comédia trash", entre outros gêneros mais despojados em relação à produção, que servem apenas para causar riso e assustar, possuindo baixos custos e tramas não tão trabalhadas.

Outra variável que influencia nas notas dos filmes, mas não teve um gráfico plotado, são os gêneros. Os gêneros de cada filme impactam nas notas já que há gêneros que o público tem mais pareço por, como por exemplo, ação, aventura, comédia.

Além do já falado acima, observamos que a classificação indicativa tem uma certa relação com a nota que é dada para cada filme. Filmes classificados em R e PG-13, são os que possuem um maior número de notas Boas e Neutras. Os filmes R são os que têm mais avaliações Muito Bom. E os PG-13 são os que possuem um maior número de avaliações Ruim. Dessa forma, pode-se ver que a classificação indicativa de cada filme tem uma certa influência na nota que eles irão receber.

As variáveis, ator 1, 2 e 3 e o diretor, também estão sendo consideradas já que as mesmas causam, talvez, a maior influência na nota de um filme, sendo estes o "core" de qualquer filme. Atores mundialmente conhecidos, que já receberam oscar estão presentes em filmes com uma nota alta, como por exemplo, Meryl Streep, Tom Hanks, Leonardo DiCaprio, etc…

Por fim, podemos concluir que para atingirmos uma noção mais exata do que realmente influencia a nota de um filme será necessário considerar todos esses indicadores juntos, já que essas variáveis não são independentes e o seu conjunto é o que compõem a nota dos filmes.

Acabada a limpeza dos dados, tivemos que criar um dataset separado de gêneros, já que os mesmos estavam grudados por uma “|”, sem espaço entre eles.

Com isso realizado, fomos capazes de criar uma grande lista, com todas as variáveis de string e passa-la no Count Vectorizer e pelo toarray, transformando a lista de strings em uma matriz de 0 e 1. Com isso, concatenamos as variáveis numéricas na matriz.

Dividimos essa matriz em base de treinamento e teste, 80% e 20% respectivamente. A primeira é usada para treinar os classificadores, e a segunda para testá-los.

Utilizamos três classificadores diferentes, para ver qual modelo teria o melhor resultado. Os escolhidos foram Multinomial Naive Bayes, Random Forest Classifier e Support Vector Classification (SVC).

Os Modelos:

- *Multinomial Naive Bayes:*

O Multinomial NB é um classificador scikit learn, o qual é uma biblioteca de machine-learning da linguagem de código python que classifica recursos discretos. Ele é utilizado na maioria das vezes para classificar dados que estão em formato de texto. Além disso, esse classificador utiliza contagens de números inteiros, como 1 e 0, mas há alguns casos que ele funciona com valores tf-idf, uma medida estatística que tem o intuito de indicar a importância de uma palavra de um documento em relação a uma coleção de documentos ou em um corpus linguístico.

Este utiliza o algoritmo de Naive Bayes para classificar uma base de dados distribuída multinomialmente, ou seja, os dados podem trazer mais de dois resultados possíveis. Ele utiliza uma matriz de vetores formada a partir dos textos da base de dados explorada. Essa matriz é composta de 1 e 0, sendo 1 quando alguma palavra tem aparição e 0 quando ela não aparece.

Assim, a distribuição é parametrizada pelos vetores que foram passados ao classificador, o qual vai observar o número de características que existem, o número de vezes que cada uma delas

aparece e o total de características que há na base de dados. O parâmetro final é estimado a partir de uma versão suavizada de máxima verossimilhança entre os dados analisados.

Então, no nosso caso, estamos pegando a maior parte dos nossos dados, os quais são textos, “vetorizamos” e transformamos-os em uma enorme matriz. Adicionamos a essa matriz as características que não são texto, como o número de críticas por usuário e aplicamos isso à biblioteca do python. Então dividimos isso em uma base de treinamento e uma de teste, 80% e 20% respectivamente. Passamos a primeira no modelo para o mesmo aprender e então passamos a base de teste, para ver quanto o modelo acertaria. No caso do Multinomial NB, recebemos um resultado muito baixo, com uma taxa de acertos de apenas 5%. O motivo de isso ocorrer deve-se ao fato do algoritmo Naive Bayes considerar todas as características independentes umas das outras, ou seja, ele não analisa elas em conjunto para chegar em uma resposta. Devido a isso, uma hipótese que temos é que, o modelo não foi capaz de ter uma alta taxa de acertos, pois no nosso caso, a junção de diversas características analisadas conjuntamente é o essencial para dar a nota do filme, e o Naive Bayes, como mencionado anteriormente, não consegue identificar isso. Um exemplo seria que, um filme dirigido por Darren Aronofsky, diretor de Cisne Negro, tendo como principais atores Natalie Portman, Meryl Streep e Hugh Jackman, é um filme com altas probabilidades de ter uma nota alta, mas não por causa de cada ator, individualmente, mas pelo que o conjunto entre esses atores e diretor pode gerar.

- Random Forest Classifier:

O Random Forest Classifier é um classificador que utiliza um sistema de árvore de decisão, ou seja, ele vai criando “galhos”, sendo cada um deles formados depois de uma decisão tomada. Ele vai dividindo a amostra principal, em diversas subamostras do conjunto de dados. Assim, o modelo vai criando uma grande rede de decisões e amostras (motivo de ser chamado de árvore de decisão), da qual o Random Forest irá usar a média, das subamostras e das decisões, para ter a melhor precisão do modelo.

Vale dizer também que a cada divisão, devido a tomada de decisão, ele não é a melhor decisão entre todas as amostras, mas sim, entre aqueles subconjuntos de amostras.

Devido ao modelo ser um pouco aleatório (Random), ele tende a aumentar o viés, ou seja, tende a sair um pouco da trajetória da resposta mais adequada, entretanto, devido ao uso da média, a variância da resposta diminui, compensando esse desvio, e tornando esse modelo, em geral, melhor que outros.

Então, neste modelo, estamos passando uma matriz, a qual possuiu todas as características de cada filme. Ao passarmos o conjunto de teste, para observar o quanto ele pode acertar, este deve ir analisando os dados e decidindo em qual categoria a informação se encaixa, por exemplo, ele analisou X ator, Y diretor e decidiu que eles eram Muito Bom, tornando essa uma melhor decisão do que dizer que esse conjunto era Neutro.

Após analisar toda a base de dados, o modelo deve ter realizado a média de suas decisões sobre cada subamostra, e devolvido a resposta sobre a categoria de cada filme. Esse modelo teve uma acurácia de 58-63% de acerto da base de teste que passamos para ele. Uma hipótese é de

que o modelo, por causa da média que ele realiza, teve uma maior porcentagem de acerto do que poderia ser. A acurácia poderia ser menor, caso isso não ocorresse.

- Support Vector Classification (SVC).

O Support Vector Classification ou SVC é um algoritmo supervisionado de Machine Learning usado para desafios de classificação e baseado no libsvm, uma biblioteca para SVMs (Support Vector Machines). Nele, plotamos cada dado como um ponto em um espaço com n dimensões com o valor de cada atributo existente sendo o valor de uma coordenada em particular. Então, para fazer o processo de classificação, acha-se um hiperplano que separa e agrega as diferentes classes.

O modelo funciona com datasets menores, porém ele é muito mais poderoso do que outros métodos. Assim, para um número de amostras com um conjuntos de dados com mais 10.000 amostras o modelo já não funciona tão bem.

Como outros classificadores, o SVC recebe como entrada duas matrizes: uma matriz X contendo as amostras de treinamento e uma matriz Y de rótulos de classe (strings ou int) tendo como tamanho o número de amostras do dataset.

Em nosso caso as duas matrizes passadas foram uma com as características e outra com o resultado de cada filme. Sendo assim, parte delas foram usadas para treinar o modelo e outra parte para verificar se o modelo aprendeu sozinho. Com isto, este método nos retornou uma boa acuracia, com 60% de acertos.

Aperfeiçoando os modelos e futuras iterações:

Na tentativa de aperfeiçoar o resultado, tentou-se duas coisas. A primeira foi utilizar um dataset muito maior, o qual continha muito mais dados do que o utilizado, o que possibilitaria uma maior base de teste, a qual levaria a uma maior taxa de acertos do modelo. Entretanto, a outra base de dados, estava totalmente sem organização e precisaria de muita atenção para então ser utilizada no projeto. O integrante Bruno, tentou melhorar o dataset o máximo possível para torná-lo usável, porém devido ao curto prazo de tempo, não fomos capazes de utilizar esse segundo dataset. Sugerimos como futura iteração continuar trabalhando nessa base de dados que tem mais de 9 milhões de filmes.

A outra tentativa de aprimorar o resultado foi colocar o underline (“_”) entre todas as strings (nomes de atores e diretores) que estavam separadas por espaço. Isso foi realizado, pois imaginávamos que o resultado gerado pelo modelo Multinomial Naive Bayes, de 5% de acertos, ocorria pelo fato de, durante o processo, todos os nomes e sobrenomes serem separados, o que influenciava no nome de cada membro do elenco, impedindo o modelo de alcançar uma bom resultado. Entretanto, descobrimos que essa não era a causa, pois mesmo após ter realizado essa alteração, o resultado do modelo não teve alteração alguma.