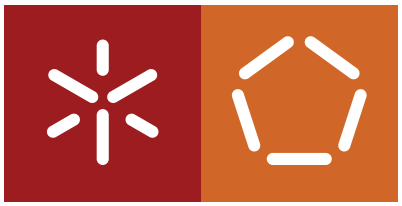
2016/2017



Universidade do Minho

Mestrado Integrado em Engenharia Informática

Business Intelligence

Análise de Dados

IMDB 5000 Movie Dataset

Gil Gonçalves - A67738
Luis Paulo Ferreira Pedro - A70415
José Pedro Santos Monteiro - A73014
Bruno Manuel Gonçalves Ribeiro - A73269

18 de Junho de 2017

Conteúdo

1 Introdução

1.1 Contextualização

Com este trabalho pretendemos implementar de um sistema de aprendizagem automática, seguindo a metodologia CRISP-DM. A metodologia CRISP-DM tem como primeira fase o estudo do negócio (do Inglês Business Understanding), que tem como função entender o que o cliente pretende realizar numa perspetiva de negócio. Na segunda fase está presente o estudo dos dados (do Inglês Data understanding), onde é realizada a aquisição e familiarização dos dados . Numa terceira etapa, é executada a preparação dos dados (do Inglês Data preparation), que é referente à decisão sobre o volume e o tipo de dados a usar no processo de análise, seguindo-se de modelação, avaliação e implementação. As seis fases foram pensadas para que pudessem ser aplicadas em qualquer área de negócio [CRISP-DM, 1999]. Neste primeiro pré relatório vamos nos focar nas primeiras 3 fases. O Internet Movie Database (também conhecido pelo acrónimo IMDb) é uma base de dados online de informação sobre música, cinema, filmes, programas e comerciais para televisão e jogos de computador. O nosso caso de estudo será baseado nos filmes e o *dataset* poderá ser encontrado em `https://www.kaggle.com/deepmatrix/imdb-5000-movie-dataset`. Este *dataset* contém 28 atributos descrevendo cerca de 5000 filmes.

1.2 Motivação e Objectivos

Temos como principal objetivo explorar o *dataset* que nos foi fornecido que contém informação de 5000 filmes, ao qual pretendemos então encontrar padrões consistentes e/ou relacionamentos sistemáticos entre variáveis e, então, validar-los aplicando os padrões detetados a novos subconjuntos de dados. O processo consiste basicamente em 3 etapas: exploração, construção de modelo ou definição do padrão e validação/verificação. Com a execução destas etapas esperamos obter uma "ferramenta"para uso futuro e assim obtermos uma noção clara da industria cinematográfica. Usar técnicas de modelação tem como objetivo prever quais as futuros filmes que irão ter sucesso tanto em questão monetárias como em questão de classificação, tendo como objetivo com esta previsão conseguir utilizar estes dados para otimizar o negócio da produção de filmes.

2 Estudo do Negócio

A ideia principal é quantificar a trajetória e a experiência das pessoas por trás dos filmes e, em seguida, aplicar a métrica resultante para estimar o sucesso dos filmes que estão em fase de pós-produção. Assim sendo, precisamos de definir uma métrica para medir o sucesso de um filme. Em particular, estamos em primeiro lugar a considerar duas áreas principais:

- Prever se o filme vai ser bem recebido pela critica (imdb raking).
- Prever se o filme vai ser um sucesso a nível monetário.

A primeira foca-se no sucesso de qualidade, enquanto que a segunda se foca no sucesso a nível de bilheteira. Para ambas penámos à partida utilizar modelos de classificação e regressão, ou seja, métodos de previsão, com o objetivo de prever com base no *dataset* se o filme será um sucesso ou não.

Durante o projeto tencionamos também responder a outras questões, nomeadamente:

- Encontrar grupos de filmes com impacto semelhante;
- Perceber se os atores e os realizadores influenciam a classificação de um filme;
- Perceber se os atores e os realizadores influenciam o lucro de um filme.

Em relação a estas últimas questões, pensámos em aplicar métodos de descrição. Na primeira questão pretendemos aplicar modelos de *clustering* de modo a agrupar os filmes em relação a atributos relevantes, de forma a tentar perceber que variáveis condicionam o impacto de um filme. Quanto ao últimos dois pontos, pensámos em aplicar modelos de *association rules*, a fim de perceber a importância dos diferentes intervenientes para o sucesso/insucesso de um filme, tanto a nível de bilheteiro como em relação ao *score* do filme.

3 Estudo dos Dados

Tendo como base as questões anteriormente explicitadas, é necessário então proceder à análise dos dados que temos a nosso dispor, a fim de posteriormente estarem preparados para lhes serem aplicados técnicas de *data mining*.

Os dados que irão ser utilizados foram retirados do site Internet Movie Database (IMDB) e encontram-se disponíveis no Kaggle. Este *dataset* é constituído por cerca de 5000 filmes e contém 28 atributos que descrevem cada filme em questão, desde o seu titulo até aos atores presentes neste. De seguida, é apresentada uma tabela que descreve cada um dos atributos presentes no *dataset*.

Atributo	Descrição	Tipo	Dominio
color	Descreve se o filme é a cores ou a preto e branco	String	Color or Black and White
director_name	Identifica quem foi o realizador do filme	String	Ex: Christopher Nolan
num_critic_for_reviews	Número de análises feitas pelos críticos	Int	1-813
duration	Duração total do filme	Int	7-511
director_facebook_likes	Quantifica o número de likes no facebook do realizador	Int	0-23000
actor_3_facebook_likes	Quantifica o número de likes no facebook do terceiro ator (grau de importância no filme).	Int	0-23000
actor_2_name	Representa o nome do segundo ator do filme (grau de importância no filme).	String	Ex:Daniel Radcliffe
actor_1_facebook_likes	Quantifica o número de likes do ator principal.	Int	0-640000
gross	Quantifica o total de dinheiro adquirido com o filme	Float	162-760505847
genres	Lista os gêneros presentes naquele filme	String	Ex:Action Adventure Thriller
actor_1_name	Representa o nome do ator principal	String	Ex: Johnny Depp

Atributo	Descrição	Tipo	Dominio
movie_title	Identifica o Nome do filme	String	Ex:Avatar
num_voted_users	Número de votos efetuados na plataforma, sendo esses votos que levam à classificação do filme.	Int	5-1689764
cast_total_facebook_likes	Número total de gostos no facebook do elenco	Int	0-656730
actor_3_name	Nome do terceiro ator (grau de importância no filme).	String	Ex:Scarlett Johansson
facenumber_in_poster	Número de caras presentes no poster do filme	Int	0-43
plot_keywords	Resumo do filme por palavras chaves.	String	Ex:avatar future native paraplegic .
movie_imdb_link	Link que redireciona para a pagina do filme no IMDB	String	EEx: http://www.imdb.com /title/tt049954 9/?ref_=fn_tt_tt_1
num_user_for_reviews	Número de análises feitas pelos utilizadores	Int	1-4667
language	Linguagem do filme.	String	Ex: English
country	País onde pertence o filme.	String	Ex: USA
content_rating	Indica qual faixa etária é adequada para exibir a referida mídia.	String	Ex: Approved
budget	Dinheiro despendido na realização do filme	Float	218-12215500000
title_year	Ano de lançamento do filme	Int	1916-2016
actor_2_facebook_likes	Quantifica o número de likes no facebook do terceiro ator (grau de importância no filme).	Int	0-137000
imdb_score	Quantifica de 1 a 10 a qualidade do filme, este ranking é atribuido pelos votos dos utilizadores	Float	1.2-9.5
aspect_ratio	A relação de aspecto de uma forma geométrica é a proporção de seus tamanhos em diferentes dimensões	Float	1.2 - 5.4
movie_facebook_likes	Número de likes que o filme tem na sua página de facebook	Int	0-349000

Tabela 1: Descrição dos atributos

Através da descrição dos vários atributos acima explicitados podemos verificar que este *dataset* possui um vasto conjunto de informações referentes aos filmes, que permitem assim tirar as conclusões inerentes a este projeto. Como tal, apesar de todos estes atributos estarem a descrever os vários filmes, alguns deles não irão ser de facto necessários para o desenrolar do projeto.

Através da exploração dos dados do *dataset*, podemos verificar que vários registos possuem valores nulos em alguns dos seus atributos, como nos atributos *country* e *budget*. Outro dos problemas encontrados prende-se ao atributo textitgenres que possui uma lista de géneros em que o filme se enquadra, sendo preciso tratar deste atributo de a ser usado nos modelos.

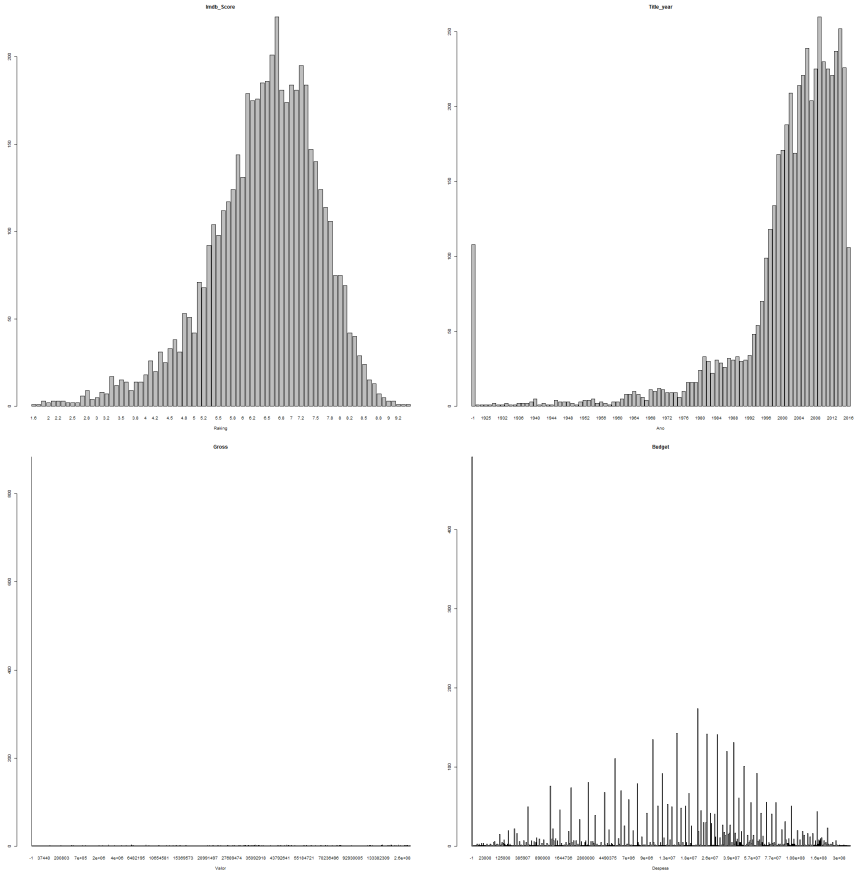


Figura 1: : Distribuição dos dados

Na figura a cima está presente a distribuição de valores dos atributos *imdb_score*,*title_year*, *gross* e *budget*. Como são valores numéricos positivos utilizou-se o valor **-1** para representar os valores desconhecidos.

Como existem valores nulos, e de forma a não se excluírem valores numéricos importante por apenas ter um campo a nulo como o nome do filme, antes de se aplicar a função *pairs* foram apenas selecionados os valores numéricos. O resultado da função *pairs* é o seguinte:

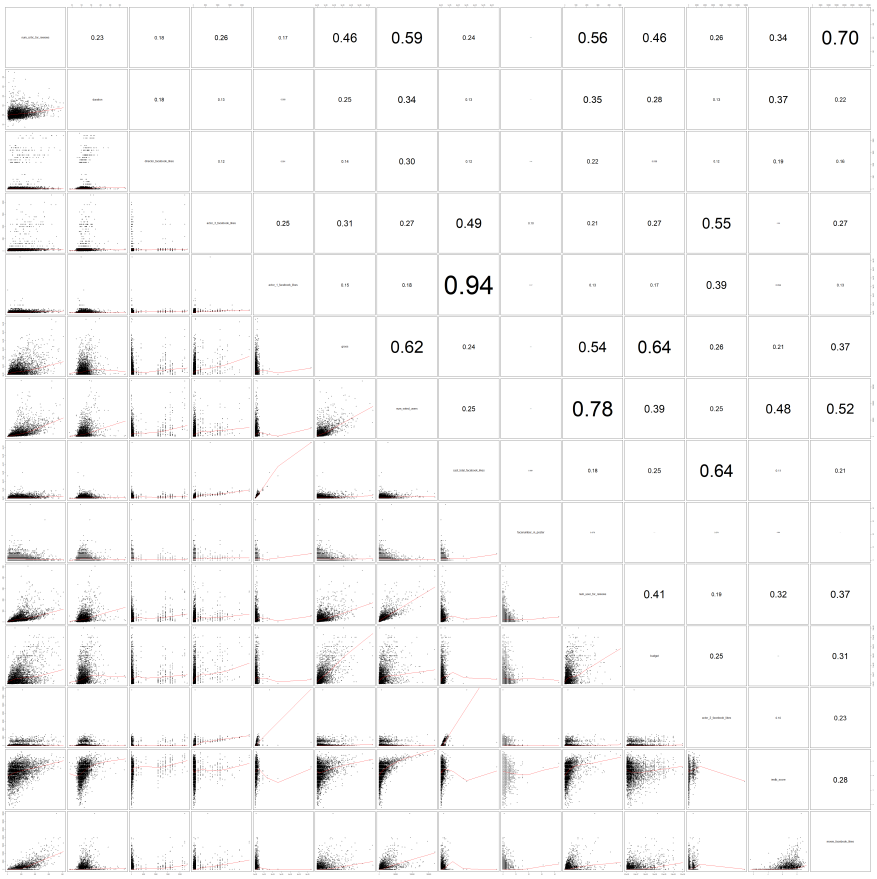


Figura 2: : Resultado da função *pairs*

Com a aplicação da função *pairs*, é possível verificar quais os atributos que mais se relacionam. Sendo que estes resultados servem para dar uma ideia sobre quais os atributos que mais se relacionam entre si, devem sempre ser olhados com uma posição critica de forma a inferir se estes fazem ou não sentido. Posteriormente, irá ser feita uma análise para verificar se as relações entre os atributos serão ou não importantes no desenvolvimento dos modelos.

4 Preparação dos Dados

Neste etapa do projeto irá ser feita uma preparação dos dados recolhidos de forma a podemos utilizar os dados nos modelos de *data mining*. Sendo que a qualidade dos dados é um fator relevante para a construção dos modelos.

Ao longo deste capítulo vamos demonstrar de que forma é que se tratou os dados, para se conseguir uma boa qualidade dos dados.

4.1 Seleção dos Dados

Em relação à seleção dos dados, nem todos os filmes podem ser selecionados para os nossos modelos por causa de possuírem atributos com valor nulo. Assim, apenas os registos que puderem ser tratados em casos de anomalias irão ser utilizados.

Com base nas questões inerentes a este projeto, verifica-se que existem atributos que não serão necessários para as análises que queremos efetuar, e como tal devem ser descartados os seguintes:

Atributo
color
facenumber_in_poster
plot_keywords
movie_imdb_link
aspect_ratio

Tabela 2: Lista de atributos eliminados.

4.2 Limpeza dos Dados

Os principais problemas encontrados em relação aos atributos foi a existência de campos com valores nulos. Se estivermos a utilizar campos como o *budget* ou o *gross* e se estes tiverem valores nulos, não é possível utilizar esses registos nos nossos modelos, pois fazer uma estimativa para estes campos introduziria erro nestes. No entanto, se for possível estimar valores para alguns campos sem afetar a correção do modelo, esses registos poderão ser usados. De seguida é apresentada a lista de atributos que possuem campos de valor nulo:

Atributo
actor_1_facebook_likes
actor_1_name
actor_2_facebook_likes
actor_2_name
actor_3_facebook_likes
actor_3_name
budget
director_name
country
director_facebook_likes
gross
language
num_critic_for_reviews
num_users_for_reviews
title_year
imdb_score

Tabela 3: Lista de atributos com campos nulos.

Outro caso que se deve ter em conta é a unidade monetária dos campos *gross* e *budget*, na qual o valor destes campos está expresso na unidade monetária do pais correspondente ao filme. Assim, como os Estados Unidos é o pais mais frequente, iremos proceder à conversão da moeda para dólares. No entanto, é preciso ter em conta que por vezes o campo *country* é nulo, pelo que é necessário aquando da criação dos modelos verificar se a utilização destes valores mesmo sem a conversão da moeda afetam negativamente o modelo, e em caso afirmativo descarta-se o registo.

4.3 Construção de Novos Dados

O atributo *genres* contém a informação dos vários géneros em que está inserido o filme separados por um delimitador. De forma a podermos usar os vários géneros de cada filme nos nossos modelos, optámos por separar este atributo por vários atributos género.

Adicionalmente, criámos também um novo atributo chamado *revenue*, que contém o lucro obtido em cada filmes, subtraindo o ao *gross* as despesas na realização do filme *budget*.

5 Modelação

Após feita toda a preparação de todos os dados, o próximo passo prende-se com a construção dos modelos propriamente ditos, a fim de procurar respostas para todas as análises que pretendemos fazer. Assim, irá ser explicitado para cada análise que nos propomos a elaborar que métodos foram usados, previsão ou de descrição, e que modelos foram aplicadas.

5.1 Encontrar filmes com impacto semelhantes

O principal objetivo desta análise, como anteriormente explicitado, pretende-se encontrar filmes cujo impacto é semelhante, em termos de dinheiro ganho com filme, dinheiro gasto na realização, entre outros, a fim de posteriormente com a análise dos resultados tentar perceber quais as variáveis que estão mais relacionadas com o sucesso ou insucesso de um dado filme.

Para esta análise, o método mais indicado é o *clustering*, visto que queremos encontrar grupos de filmes com características semelhantes. O modelo que iremos usar será o *kmeans*, ao qual tivemos de ter especial cuidado com a normalização dos atributos e com a atribuição do número de *clusters*.

Assim, foi feita uma seleção inicial dos atributos necessários para a realização desta análise, isto com base na correlação dos mesmos, já após um filtro inicial de alguns *outliers* que estavam a dificultar o processo de análise. Para a remoção dos Os atribu*, o processo foi baseado na análise da dispersão dos atributos gross e budget, na qual removemos os que apresentavam valores muito distantes dos restantes, através da mediana. outliers*tos selecionamos foram os seguintes: *num_critic_for_reviews*, *num_voted_users*, *num_user_for_reviews*, *movie_facebook_likes*, *gross*, *budget*, *imdb_score* e *revenue*. Outros atributos que pensávamos que seriam importantes para esta análise, como o número de *likes* das páginas do Facebook do diretor e dos atores, demonstraram-se através do modelo que não apresentavam uma forte indicação do impacto no *clustering* dos filmes, pelo que os descartamos da análise.

Para a construção do modelo foram usados a totalidade dos dados após o processo de seleção e a filtragem de alguns *outliers*. Para avaliar o sucesso/insucesso do modelo, utilizámos como critério a facilidade de interpretação do mesmo, pelo que assim conseguimos filtrar atributos que entendemos não serem necessários.

Estando agora preparados para correr o modelo, inicialmente fizemos um teste para determinar qual o valor correto para número de *clusters* a usar, de forma a melhorar o modelo. De seguida é apresentado o gráfico com o cálculo da distribuição da soma dos quadrados.

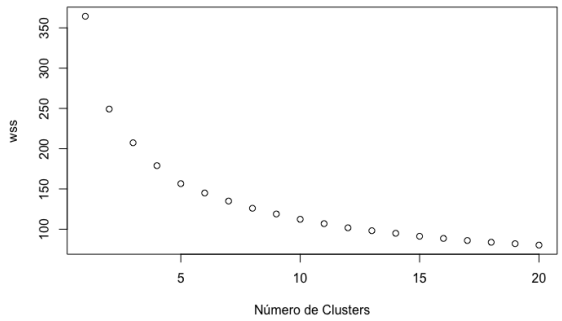


Figura 3: Relação entre a distribuição da soma dos quadrados e o número de *clusters*.

Com base neste gráfico, podemos concluir que o erro diminuiu substancialmente até ao número de 3, pelo que a partir desse momento a diminuição ocorre praticamente constante. No entanto, um maior número de *clusters* poderiam ser usados, mas tornavam a interpretação do mesmo mais complicado. Note que fixámos o parâmetro `nstart` a 25.

De seguida, é apresentado os resultados do modelo.

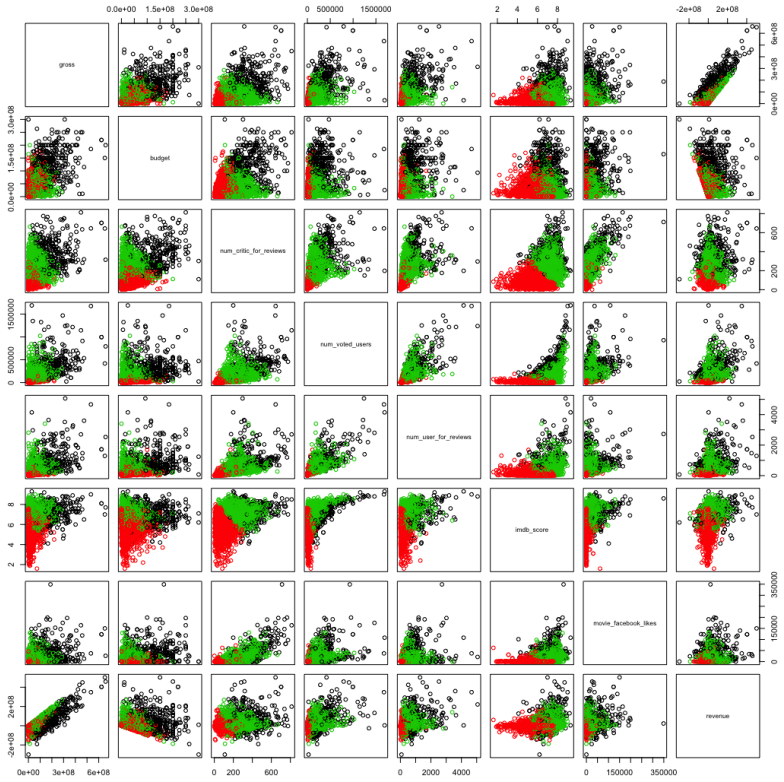


Figura 4: Rsultado da aplicação do modelo de *clustering*.

Uma avaliação mais completa do modelo irá ser apresentada no capítulo posterior. No entanto, podemos desde já adiantar que conseguimos concluir, por exemplo, filmes com grandes custos de realização (*budget*) levam a *scores* de IMDB mais altos. Contudo, a qualidade dos dados afetou negativamente o modelo, muito devido ao facto destes serem muito heterogéneos, mesmo após a fase de tratamento dos mesmos.

5.2 Modelo de previsão para o IMDB score de um filme

Sendo que o objetivo é criar um modelo de previsão para o IMDB *score* de um filme, recorreremos ao método de regressão, ao qual implementámos os modelos de regressão linear, *decision trees* e *random forest*.

Já com os dados preparados para utilizar (normalizados), fizemos uma seleção inicial daqueles que achámos mais relevantes para o problema em questão. Os atributos que utilizamos foram os seguintes: *imdb_score*, *budget*, *movie_facebook_likes*, *duration*, *director_facebook_likes*. Outros atributos foram também utilizados numa fase de avaliação da performance dos vários modelos, mas tinham um impacto negativo, pelo que foram removidas. Outros atributos, como o *gross* ou *num_voted_users*, não foram usados pois, nestes casos, são obtidos após o filme ter sido lançado, sendo valores que à priori não sabemos.

É importante referir que os dados foram divididos em dois conjuntos, de treino e de teste, sendo que têm 70% e 30% dos dados, respetivamente. Denote também que para o modelo *random forest*

foram usados os parâmetros *default* para as variáveis *mtry* e *ntree*, pois com base nos resultados de performance concluímos que o modelo se comportava melhor com esses valores.

Por fim, foi necessário definir um conjunto de métricas de forma a avaliar a performance dos modelos. As métricas usadas foram a RMSE e R2, que correspondem à *root mean squared error* e *coefficient of determination*, respetivamente. Para a métrica RMSE, no melhor caso, o valor obtido deve ser 0, sendo que para a métrica R2 deve ser 1. De seguida, é apresentada uma tabela com os resultados de cada uma das métricas obtidos para cada modelo.

Modelo	RMSE	R2
Regressão Linear	0.037	0.204
Decision Tree	0.055	0.236
Random Forest	0.026	0.363

Tabela 4: Resultados obtidos para as métricas RMSE e R2.

Através da tabela acima apresentada, podemos concluir que o modelo *random forest* apresentou melhores resultados que os restantes. Uma melhor avaliação do modelo irá ser apresentada no capítulo posterior.

5.3 Modelo de previsão para a receita bruta de um filme

Analogamente à análise anterior, visto que queremos criar um método de previsão para a receita de um dado filmes, utilizamos o método de regressão, ao qual implementámos também os três modelos anteriormente explicitados.

Todo o processo desenvolvido na construção do modelo de previsão para o IMDB *score* foi utilizado também para a construção deste modelo, pelo que a única diferença resumiu-se à utilização de diferentes atributos para os modelos.

Neste caso, os atributos utilizados foram: *gross*, *budget*, *imdb_score*, *num_critic_for_reviews*, *num_voted_users*, *num_user_for_reviews*. Apenas foram usadas variáveis que favoreciam a performance dos modelos, na qual usámos as métricas anteriormente explicitadas para a avaliação dos modelos em questão (RMSE e R2). De seguida, é apresentada uma tabela com os resultados obtidos para os modelos utilizados:

Modelo	RMSE	R2
Regressão Linear	0.548	-15.019
Decision Tree	0.549	-15.057
Random Forest	0.550	-15.089

Tabela 5: Resultados obtidos para as métricas RMSE e R2.

Com base nos resultados obtidos, podemos concluir que todos os modelos tiveram uma performance muito semelhante, pelo que qualquer um dos modelos pode ser utilizado para a previsão da receita de filmes. No capítulo posterior irá ser feita uma avaliação ao modelo.

5.4 Perceber se os atores e os realizadores influenciam a classificação de um filme

Para responder a esta pergunta usou-se regras de associação, com o algoritmo *apriori* para tentar perceber quais os atores principais e realizadores que em conjunto produziam filmes que agradam aos utilizadores.

Para isso utilizou-se os atributos *director_name*, o *actor_1_name* e o *imdb_score*. A escala do *imdb_score* está compreendida entre [1.6;9.3], sendo que grande parte dos valores está compreendida

entre [6.5;7.5]. Como as regras de associação vão sempre para os atributos que apresentam maior frequência, para tentar atenuar estes resultados criou-se seis escalas:

- muito_fraco → varia entre [0;3]
- fraco → varia entre]3;6]
- medio → varia entre]6;7]
- bom → varia entre]7;7.5[
- muito_bom → varia entre [7.5;8.5]
- sucesso → varia entre]8.5;10]

e deste modo conseguiu-se separar melhor os valores para que os mesmo não se concentrassem numa só região.

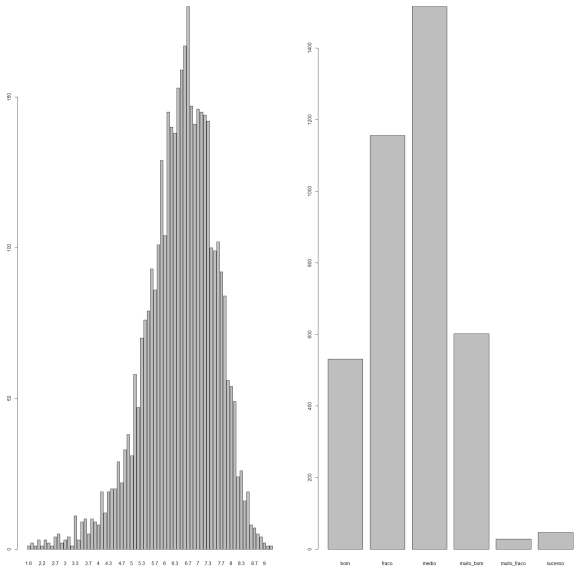


Figura 5: Comparação entre os valores em bruto com a separação por categorias.

Para a construção do modelo foram utilizados a totalidade dos atributos, após os mesmos serem passados por um processo de filtragem, retirando os valores nulos.

Após o processo de modelação de dados aplicou-se o algoritmo *apriori* nos dados com um suporte de 0.0015 com uma confiança de 60%. Foi usado um valor tão baixo no suporte devido à quantidade de dados presentes no *dataset*.

Os resultados obtidos foram os seguintes:

```
     lhs                          rhs       support     confidence lift
[1]  {Brian Levant}            => {fraco}   0.001545993 1.0000000   3.357266
[2]  {Michael Winterbottom}    => {medio}   0.001545993 1.0000000   2.560026
[3]  {Jonathan Liebesman}      => {fraco}   0.001545993 1.0000000   3.357266
[4]  {Tyler Perry}             => {fraco}   0.001545993 1.0000000   3.357266
[5]  {Raja Gosnell}            => {fraco}   0.001545993 1.0000000   3.357266
[6]  {Michael Jai White}       => {fraco}   0.001803659 1.0000000   3.357266
[7]  {Jon Turteltaub}          => {medio}   0.001803659 1.0000000   2.560026
[8]  {Jay Roach}               => {medio}   0.001803659 1.0000000   2.560026
[9]  {Peter Segal}             => {medio}   0.001545993 0.8571429   2.194308
[10] {Sarah Michelle Gellar}   => {fraco}   0.001803659 0.8750000   2.937608
[11] {Garry Marshall}          => {fraco}   0.001545993 0.7500000   2.517950
[12] {Christopher Nolan}       => {sucesso} 0.001545993 0.7500000  61.930851
[13] {Amy Poehler}             => {fraco}   0.001545993 0.7500000   2.517950
[14] {Zooey Deschanel}         => {medio}   0.001803659 0.7777778   1.991132
[15] {Robin Wright}            => {medio}   0.001545993 0.6666667   1.706684
```

Figura 6: Resultados obtidos com a confiança=60% e suporte=0.0015.

Os resultados do algoritmo mais tarde serão analisados e posteriormente avaliados para verificar se estes são aceitáveis.

5.5 Perceber se os atores e os realizadores influenciam o lucro de um filme

Para responder a esta pergunta usou-se regras de associação, com o algoritmo *apriori* para tentar perceber se os atores principais e o diretor tinham influência no lucro do filme.

Para isso utilizou-se os atributos *director_name*, o *actor_1_name*,o *gross* e o *budget* . Para se determinar o lucro criou-se um atributo chamado *Lucro* que é a diferença entre o *gross* e o *budget*. Depois percorreu-se o *dataset* e substitui-se os seguintes valores:

- Valores menor que zero → Prejuízo
- Valores igual a zero → Lucro zero
- Valores maior que zero → Lucro

deste modo é possível representar melhor os valores para uma melhor analise ao problema.

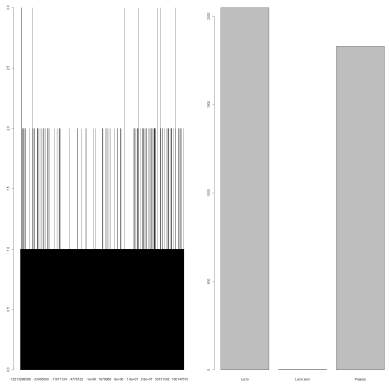


Figura 7: Comparação entre os valores do lucro em bruto com a separação por categorias.

Apesar de existir um valor com lucro zero, esse valor não vai ser relevante para o problema visto que é só um valor. Os resultados foram obtidos com o algoritmo usando uma confiança de 60% e com um suporte de 0.0015. Foi usado um valor tão baixo no suporte devido à quantidade de dados presentes no *dataset*.

Os resultados obtidos foram os seguintes:

```
     lhs                       rhs        support     confidence lift
[1]  {Michael Winterbottom} => {Prejuizo} 0.001545993 1.0000000  2.119607
[2]  {Lin Shaye}            => {Lucro}    0.001545993 1.0000000  1.894095
[3]  {Tim Story}            => {Lucro}    0.001545993 1.0000000  1.894095
[4]  {Jada Pinkett Smith}   => {Lucro}    0.001545993 1.0000000  1.894095
[5]  {Tyler Perry}          => {Lucro}    0.001545993 1.0000000  1.894095
[6]  {Andy Fickman}         => {Lucro}    0.001545993 1.0000000  1.894095
[7]  {Nia Long}             => {Lucro}    0.001545993 1.0000000  1.894095
[8]  {Catherine Deneuve}    => {Prejuizo} 0.001545993 1.0000000  2.119607
[9]  {Alyson Hannigan}      => {Lucro}    0.001803659 1.0000000  1.894095
[10] {James Wan}            => {Lucro}    0.001545993 0.8571429  1.623510
```

Figura 8: Resultados obtidos com a confiança=60% e suporte=0.0015

Os resultados do algoritmo mais tarde serão analisados e posteriormente avaliados para ver se são aceitáveis.

6 Avaliação

Ao longo desta secção vamos proceder a uma avaliação individual de cada um dos modelos que criámos, explicitando pontos positivos e negativos, e também novas abordagens para contornar os problemas identificados.

6.1 Encontrar filmes com impacto semelhantes

Os resultados obtidos para este modelo não foram satisfatórios, algo que podemos verificar pela qualidade na divisão dos *clusters*. No entanto, através da análise da dispersão dos valores, podemos ainda assim tirar algumas conclusões.

Podemos então concluir que o IMDB *score* está relacionado os atributos explicitados, sendo que apenas não acontecem casos em que uma classificação baixa leva a, por exemplo, um *gross* alto, ou então a elevadas despesas de realização (*budget*). Um dado interessante é que existem muitos casos em que os filmes não têm lucro, mesmo em casos em que o IMDB *score* é alto.

No entanto, o modelo de *clustering* poderia apresentar melhores resultados, algo que se pode afirmar devido à precisão com que os *clusters* então definidos, pois é notório que estes se apresentam sobrepostos. Isto deveu-se à qualidade dos dados com que estávamos a trabalhar, que mesmo com tratamento de alguns *outliers*, não apresentou resultados significativamente melhores.

Esforços foram feitos no sentido de remover ainda mais os *outliers*, mas a quantidade de dados que eram filtrados era substancialmente grande que decidimos apenas ficar por uma remoção menos agressiva. Outra das estratégias foi tratar dos nulos através de estratégias regressão linear e aplicar a média, mas apenas piorava o modelo. No entanto, podem ser experimentadas outras técnicas para a remoção de *outliers* que apresentem melhores resultados, sendo algo que se deve ter em conta para futuras melhorias ao modelo.

6.2 Modelo de previsão para o IMDB score de um filme

Em relação a este modelo, os resultados obtidos foram em certa parte satisfatórios, pois as correlações com que estamos a trabalhar não eram suficientemente altas para alcançar uma boa performance.

Em relação às duas métricas utilizadas, a RMSE deste modelo mostrou-se muito próximo do 0, o que é um bom indicador. No entanto, a R2 ficou ainda distanciada do 1, ficando-se pelos 0.363. Com base nos dados que estávamos a usar, estes indicadores apresentaram resultados razoáveis, mas, no entanto, era necessário melhorar o modelo, começando logo por um melhor tratamento dos dados.

A qualidade dos dados afetou negativamente o modelo, e apesar do tratamento de *outliers*, que melhorou significativamente os resultados, passando de valores de R2 de 0.127 para 0.363, seria necessário um tratamento mais severo dos mesmos. No entanto, este modelo revelou-se melhor que um classificador aleatório, o que já é um ponto positivo.

6.3 Modelo de previsão para a receita bruta de um filme

Já em relação a este modelo, devido à semelhança de construção com o modelo anterior, as conclusões a retirar são as mesmas. Com base nas correlações com que estávamos a trabalhar, o modelo apresentou resultados razoáveis.

Em relação às métricas utilizadas, o RMSE fixou-se nos 0.548, e o R2 nos -15.019. Estes valores estão aquém das expectativas, já que os valores de correlação com que estávamos a trabalhar são mais elevados em relação ao modelo anterior. No entanto, a variável que estávamos a prever era o *gross*, pelo que por causa da sua natureza e dispersão, mesmo aplicando uma normalização, é normal que estes valores sejam obtidos.

Neste caso, uma melhor alternativa seria aplicar um método de classificação, sendo preciso categorizar o atributo *gross* em categorias que iam desde o nível baixo ao nível alto. Assim, conseguiríamos

resolver o problema dos *outliers*, pelo que podemos inferir que teríamos um modelo mais preciso. No entanto, este modelo revelou-se melhor que um classificador aleatório, o que já é um ponto positivo.

6.4 Perceber se os atores e os realizadores influenciam a pontuação dada ao filme

Consoante os resultados obtidos na fase de modelação, é notório que a combinação de certo atores influenciam o sucesso do filme a nível do *ranking*. Com o modelo aplicado não se consegui obter respostas adicionais que fossem úteis para o negócio tratado. A meio do percurso desta fase foi se testando vários suportes, para que fosse possível relacionar os realizadores com o ator principal e obter a classificação do filme, devido ao grande volume de dados na qual o número de vezes em que o realizador se relaciona com o ator é de baixa frequência.

Uma surpresa agradável foi a não ocorrência de *dead ends*, o que facilitou o processo de modelagem. Por outro lado, foi notória a ocorrência de situações de repetição de regras com diferentes graus de confiança. Um exemplo é apresentado na seguinte imagem:

```
[1128] {Jamie Lee Curtis,
        John Carpenter}          => {muito_bom}          0.0007729967  0.6000000    3.868106
[1129] {Jamie Lee Curtis,
        muito_bom}               => {John Carpenter}     0.0007729967  1.0000000  298.538462
```

Figura 9: Exemplo de um caso repetido.

O exemplo apresentado acima não é considerado um erro, pois tendo em conta o baixo nível de suporte, o algoritmo tenta mais níveis de permutações o que irá originar mais regras e consequentemente algumas das regras podem aparecer repetidas, ou seja, com a mesma lógica.

6.5 Perceber se os atores e os realizadores influenciam a o lucro de um filme

Analogamente ao caso de estudo referido anteriormente, o respetivo processo de modelagem foi bastante semelhante e os problemas que surgiram foram também equivalentes. principal diferença foi que neste processo foram utilizados os atributos *gross* e *budget* de modo a calcular para cada registo o lucro de cada filme.

Posteriormente, ao aplicar o modelo de *association rules*, alcançou-se então resultados bastantes claros conseguindo assim verificar que existe um vasto número de combinações de atores e diretores. Neste caso, existe a diferença em relação ao problema anterior na qual os valores estão mais balanceados, apresentando desta forma melhores resultados.

7 Implementação

Tendo como base todo o processo anteriormente explicitado, chegou a altura de analisar os modelos desenvolvidos e encontrar formas de utilizar as conclusões obtidas para resolver/melhorar os problemas existentes e compreender o que poderá ser realmente aplicado ao contexto do negócio.

Com base nos resultados fornecidos pelos modelos, verificámos que será necessário melhorar estes, mais concretamente os modelos de previsão implementados. Estes apresentam resultados razoáveis, mas seria necessário um modelo mais robusto para justificar a sua aplicabilidade em novos contextos. Aumentar a robustez destes modelos implicará uma melhor compreensão e uma análise mais profunda dos atributos, que passará por aplicar um melhor processo de filtragem dos *outliers*. O mesmo se aplica ao modelo de *clustering* desenvolvido, na qual um melhor processo de filtragem poderá conduzir à melhoria do mesmo.

Em relação aos modelos de *association rules*, estes mostraram-se com potencial para futuras inferências, como analisar que diretores e atores escolher para garantir o sucesso de um dado filme a realizar. Assim, melhorias em relação a estes modelos mostravam-se vantagens, vistos que podem ter aplicabilidade em futuras realizações de filmes.

Seria também interessante desenvolver um sistema de recomendações de filmes com base no género, e também com a utilização do atributo *content_rating*, diretor, atores, entre outros. O género foi um atributo que não utilizámos na construção dos nossos modelos, no entanto fizemos algumas análises do mesmo, que são apresentadas no anexo A.

Por fim, para futuras melhorias aos modelos desenvolvidos e para ajudar na análise dos atributos, foram desenvolvidos gráficos com a análise de dispersão dos atributos, que poderão ser consultados no anexo B

8 Conclusão

Chegando assim o momento final do projeto prático de Análise de Dados, podemos afirmar que conseguimos aplicar os objetivos propostos para a realizam do mesmo. No entanto, foram identificadas algumas dificuldades à realização do projeto, que estão relacionadas com o tratamento dos dados do nossa dataset.

A qualidade dos dados foi um fator que complicou a implementação dos modelos, devido à existência de inúmeros outliers, ao qual mesmo após um processo inicial de filtragem dos mesmos, deparámos-nos com a redução do tamanho dos dados que tínhamos para os modelos. Assim, tivemos de aplicar um processo de filtragem menos agressivo, que melhorou as correlações entre os nossos atributos, ainda que não o suficiente para melhorar significativamente os nossos modelos. Tentámos também tratar dos valores nulos através de regressões lineares ou através da atribuição da média, mas tiveram impacto negativo nos nossos modelos desenvolvidos.

No entanto, conseguimos aplicar os modelos com sucesso para responder às questões a que nos propusemos, e também conseguimos inferir sobre qual o melhor método a aplicar para um caso específico. Seria ainda interessante explorar outras alternativas aos modelos utilizados, como por exemplo na previsão do atributo gross, na qual pensámos que seria melhor aplicar um método de classificação, categorizando assim o gross em diferentes níveis, desde baixo a alto.

Por fim, seria também interessante explorarmos melhor os géneros de filmes, verificando, por exemplo, as relações que têm uns com os outros e quais é que estão mais ligados ao sucesso/insucesso de um filme, entre outros.

A Análise em relação ao atributo género

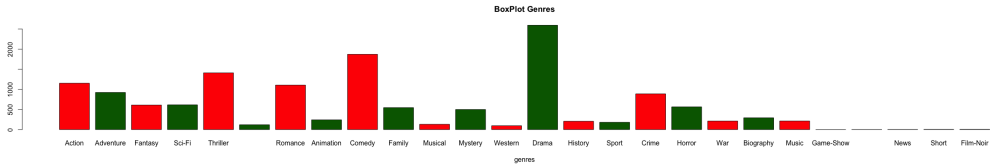


Figura 10: Distribuição de filmes por géneros.

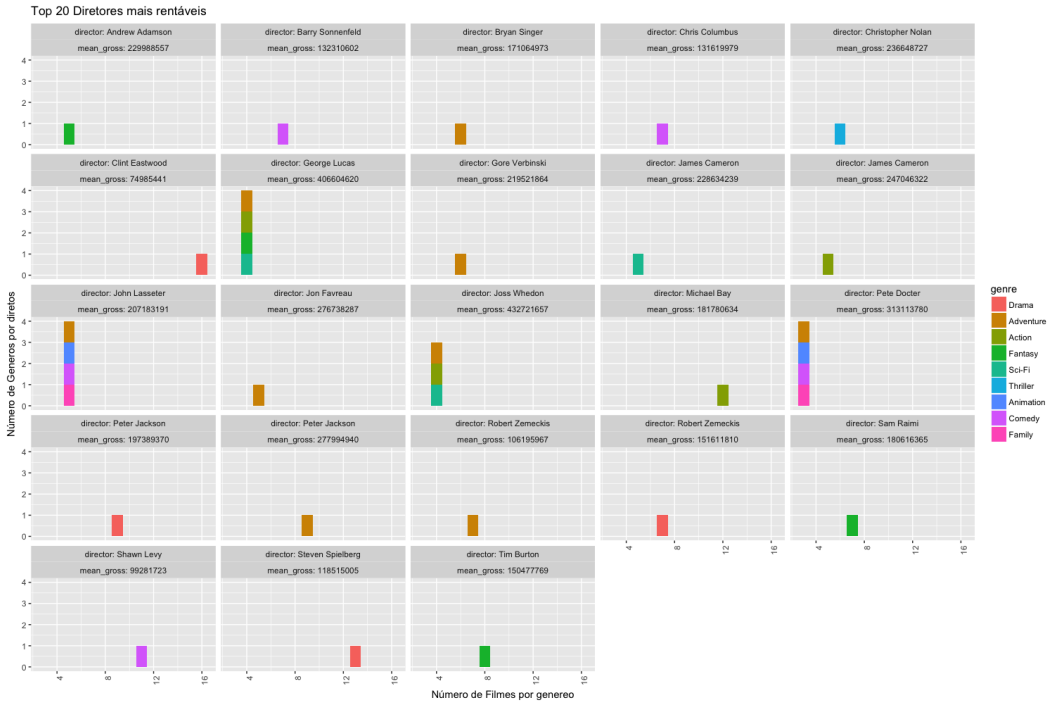
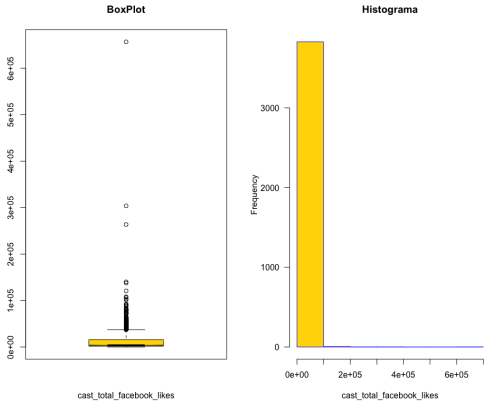
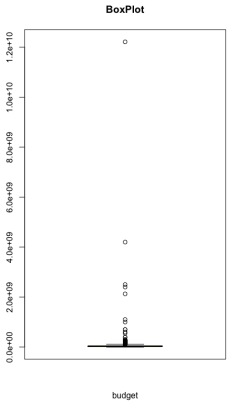
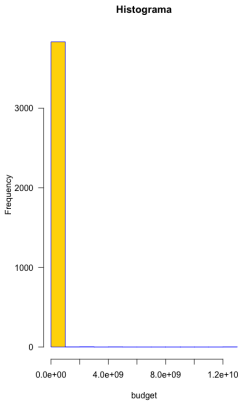


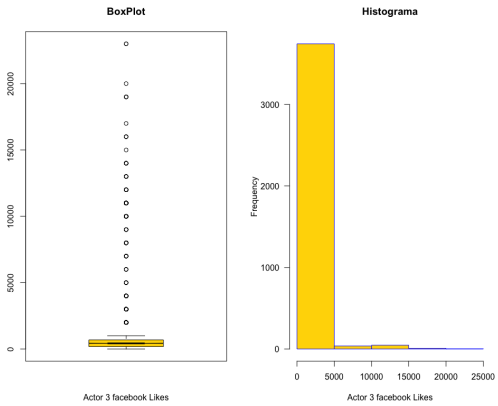
Figura 11: Top 20 diretores mais rentáveis.

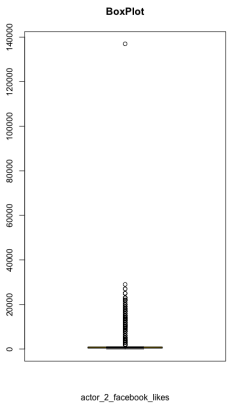
B Análise da distribuição dos dados numéricos

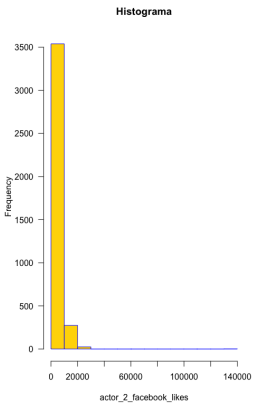


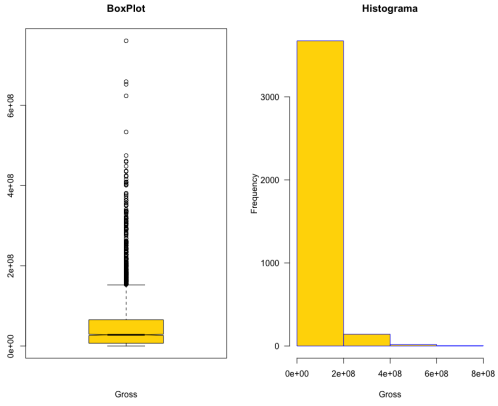


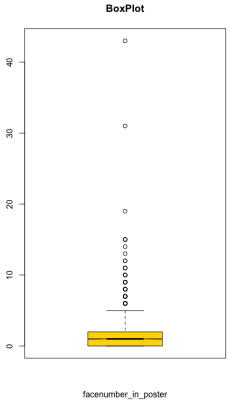


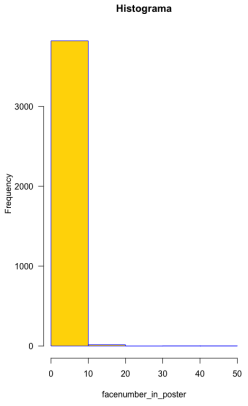


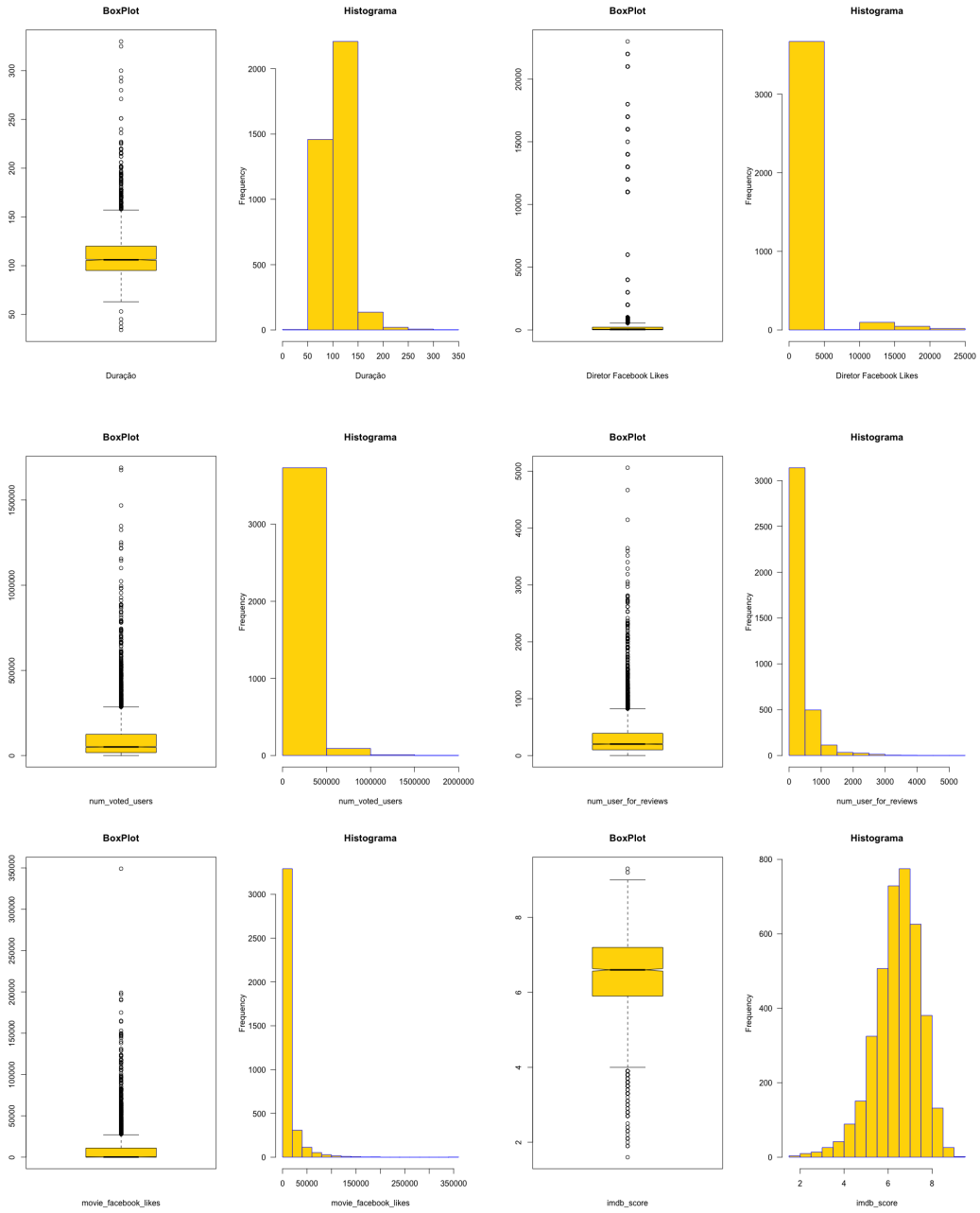


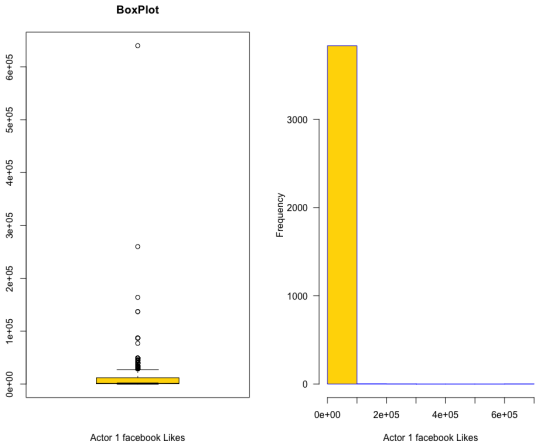









BoxPlot
Duração
Histograma
Frequency
Duração
BoxPlot
Diretor Facebook Likes
Histograma
Frequency
Diretor Facebook Likes
BoxPlot
num_voted_users
Histograma
Frequency
num_voted_users
BoxPlot
num_user_for_reviews
Histograma
Frequency
num_user_for_reviews
BoxPlot
movie_facebook_likes
Histograma
Frequency
movie_facebook_likes
BoxPlot
imdb_score
Histograma
Frequency
imdb_score


BoxPlot
0e+00
1e+05
2e+05
3e+05
4e+05
5e+05
6e+05
Actor 1 facebook Likes
Frequency
0
1000
2000
3000
0e+00
2e+05
4e+05
6e+05
Actor 1 facebook Likes

C Código R Modelo 1

```
set.seed(12345)

imdb_full <- read.csv(file = "dataset.csv", header = TRUE)

for(i in 29:length(imdb_full)) {
  imdb_full[is.na(imdb_full[i]),i]<-0
}

imdb_full <- imdb_full[complete.cases(imdb_full),]
imdb_full[imdb_full==""] <- NA

med = median(imdb_full$gross)
imdb_full <- imdb_full[imdb_full$gross < med*25,]

med = median(imdb_full$budget)
imdb_full <- imdb_full[imdb_full$budget < med*15,]

revenue <- imdb_full$gross - imdb_full$budget
imdb_full <- cbind(imdb_full, revenue)

imdb <- imdb_full[,c(9,23,3,13,19,26,28,55)]

maxs <- apply(imdb, 2, max)
mins <- apply(imdb, 2, min)
mtscaled <- as.data.frame(scale(imdb, center = mins, scale = maxs - mins))

wss <- array(dim = 20)
for(i in 1:20) {
  wss[i] = kmeans(mtscaled, centers = i, nstart = 25)$tot.withinss
}
plot(wss, xlab = "Número de Clusters")

k <- 3

model.kmeans2 <- kmeans(mtscaled, centers = k, nstart = 25)
plot(imdb, col=model.kmeans2$cluster)
points(model.kmeans2$center, col=1:2, pch=8, cex=1)
```

D Código R Modelo 2

```
library(rpart)
library(randomForest)
library(caret)

set.seed(12345)

imdb_full <- read.csv(file = "dataset.csv", header = TRUE)

for(i in 29:length(imdb_full)) {
  imdb_full[is.na(imdb_full[i]),i]<-0
}

imdb_full <- imdb_full[complete.cases(imdb_full),]
imdb_full[imdb_full==""] <- NA

med = median(imdb_full$gross)
imdb_full <- imdb_full[imdb_full$gross < med*25,]

med = median(imdb_full$budget)
imdb_full <- imdb_full[imdb_full$budget < med*15,]

revenue <- imdb_full$gross - imdb_full$budget
imdb_full <- cbind(imdb_full, revenue)

imdb <- imdb_full[,c(26,23,28,4,5)]

maxs <- apply(imdb[,-1], 2, max)
mins <- apply(imdb[,-1], 2, min)
mtscaled <- as.data.frame(scale(imdb[,-1], center = mins, scale = maxs - mins))
mtscaled <- cbind(imdb[,c("imdb_score")],mtscaled)
colnames(mtscaled)[1] <- "imdb_score"

panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...)
{
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r <- abs(cor(x, y))
    txt <- format(c(r, 0.123456789), digits = digits)[1]
    txt <- paste0(prefix, txt)
    if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
    text(0.5, 0.5, txt, cex = cex.cor * r)
}

#png("cor.png",width = 5080,height = 5080)
pairs(imdb, lower.panel = panel.smooth, upper.panel = panel.cor)
#dev.off()

train <- sample(1:nrow(mtscaled),size=ceiling(0.7*nrow(mtscaled)),replace = FALSE)
imdb.train <- mtscaled[train,]
imdb.test <- mtscaled[-train,]
```

```
formula <- imdb_score ~ budget + movie_facebook_likes + duration + director_facebook_likes

# regressao linear
lm.model <- lm(formula = formula, data = imdb.train)
lm.predict <- predict(lm.model, imdb.test[,-1])
lm.predict <- round(lm.predict,1)

lm.R2 <- 1 - ( sum((imdb.test$imdb_score - lm.predict)^2)
    / sum((imdb.test$imdb_score - mean(imdb.test$imdb_score))^2) )
lm.RMSE <- sqrt(mean(lm.predict - imdb.test$imdb_score)^2)

# decision tree
dt.model <- rpart(formula = formula, data = imdb.train)
dt.predict <- predict(dt.model, imdb.test[,-1])
dt.predict <- round(dt.predict,1)

dt.R2 <- 1 - ( sum((imdb.test$imdb_score - dt.predict)^2)
    / sum((imdb.test$imdb_score - mean(imdb.test$imdb_score))^2) )
dt.RMSE <- sqrt(mean(dt.predict - imdb.test$imdb_score)^2)

#random forest
rf.model <- randomForest(formula = formula, data = imdb.train)
rf.predict <- predict(rf.model, imdb.test[,-1])
rf.predict <- round(rf.predict,1)

rf.R2 <- 1 - ( sum((imdb.test$imdb_score - rf.predict)^2)
    / sum((imdb.test$imdb_score - mean(imdb.test$imdb_score))^2) )
rf.RMSE <- sqrt(mean(rf.predict - imdb.test$imdb_score)^2)
```

E Código R Modelo 3

```
library(rpart)
library(randomForest)

set.seed(12345)

imdb_full <- read.csv(file = "dataset.csv", header = TRUE)

for(i in 29:length(imdb_full)) {
  imdb_full[is.na(imdb_full[i]),i]<-0
}

imdb_full <- imdb_full[complete.cases(imdb_full),]
imdb_full[imdb_full==""] <- NA

med = median(imdb_full$gross)
imdb_full <- imdb_full[imdb_full$gross < med*25,]

med = median(imdb_full$budget)
imdb_full <- imdb_full[imdb_full$budget < med*15,]

revenue <- imdb_full$gross - imdb_full$budget
imdb_full <- cbind(imdb_full, revenue)

imdb <- imdb_full[,c(9,3,4,5,13,19,23,26,28)]

maxs <- apply(imdb, 2, max)
mins <- apply(imdb, 2, min)
mtscaled <- as.data.frame(scale(imdb, center = mins, scale = maxs - mins))

panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y))
  txt <- format(c(r, 0.123456789), digits = digits)[1]
  txt <- paste0(prefix, txt)
  if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r)
}

#png("cor.png",width = 5080,height = 5080)
pairs(imdb, lower.panel = panel.smooth, upper.panel = panel.cor)
#dev.off()

train <- sample(1:nrow(mtscaled),size=ceiling(0.7*nrow(mtscaled)),replace = FALSE)
imdb.train <- mtscaled[train,]
imdb.test <- mtscaled[-train,]

formula <- gross ~ budget + imdb_score + num_critic_for_reviews +
  num_voted_users + num_user_for_reviews
```

```r
# regressao linear
lm.model <- lm(formula = formula, data = imdb.train)
lm.predict <- predict(lm.model, imdb.test[,-1])

lm.R2 <- 1 - ( sum((imdb.test$imdb_score - lm.predict)^2)
    / sum((imdb.test$imdb_score - mean(imdb.test$imdb_score))^2) )
lm.RMSE <- sqrt(mean(lm.predict - imdb.test$imdb_score)^2)

# decision tree
dt.model <- rpart(formula = formula, data = imdb.train)
dt.predict <- predict(dt.model, imdb.test[,-1])

dt.R2 <- 1 - ( sum((imdb.test$imdb_score - dt.predict)^2)
    / sum((imdb.test$imdb_score - mean(imdb.test$imdb_score))^2) )
dt.RMSE <- sqrt(mean(dt.predict - imdb.test$imdb_score)^2)

#random forest
rf.model <- randomForest(formula = formula, data = imdb.train)
rf.predict <- predict(rf.model, imdb.test[,-1])

rf.R2 <- 1 - ( sum((imdb.test$imdb_score - rf.predict)^2)
    / sum((imdb.test$imdb_score - mean(imdb.test$imdb_score))^2) )
rf.RMSE <- sqrt(mean(rf.predict - imdb.test$imdb_score)^2)
```

F Código R Modelo 4

```
library(grid)
library(gdata)
library(arulesViz)
library(Matrix)
library(arules)

set.seed(123456)

ficheiro<-read.csv("outMovieFinal.csv",header = T, stringsAsFactors = TRUE)

for(i in 29:length(ficheiro)) {
  ficheiro[is.na(ficheiro[i]),i]<-0
}

ficheiro<-ficheiro[,c(-4,-16,-17,-18,-22,-27,-10)]

array<-which(ficheiro[2]=="", arr.ind = T)
ficheiro<-ficheiro[-array[,1],]
array<-which(ficheiro[1]=="", arr.ind = T)

for(i in 1:length(array[,"row"])){
  ficheiro[array[i,1],"color"]<-"Color"
}

for(i in 1:length(ficheiro)){
    array<-which(is.na(ficheiro[i]),arr.ind = T)
    if(length(array)>0) {
       ficheiro<-ficheiro[-array[,1],]
    }
}

pergunta4<-ficheiro[,c(1,2,6,9,13,15,16,18,20)]
pergunta4<-pergunta4[,-c(1,6,7,3,5,8)]

ARRAY<-which(pergunta4[3]>=8.5, arr.ind = T)

for(i in 1:nrow(pergunta4)) {
  if(pergunta4[i,"imdb_score"]>=0 && pergunta4[i,"imdb_score"]<=3 ) {
    pergunta4[i,"imdb_score"]<-"muito_fraco"
  }

  if(pergunta4[i,"imdb_score"]>3 && pergunta4[i,"imdb_score"]<=6 ) {
    pergunta4[i,"imdb_score"]<-"fraco"
  }

  if(pergunta4[i,"imdb_score"]>=6.1 && pergunta4[i,"imdb_score"]<=7 ) {
    pergunta4[i,"imdb_score"]<-"medio"
  }

  if(pergunta4[i,"imdb_score"]>7 && pergunta4[i,"imdb_score"]<7.5 ) {
```

```
    pergunta4[i,"imdb_score"]<-"bom"
  }

  if(pergunta4[i,"imdb_score"]>=7.5 && pergunta4[i,"imdb_score"]<8.5 ) {
    pergunta4[i,"imdb_score"]<-"muito_bom"
  }
}

pergunta4[ARRAY[,"row"],"imdb_score"]<-"sucesso"

par(mfrow=c(1,2))

barplot(table(ficheiro$imdb_score))
barplot(table(pergunta4$imdb_score))
min(ficheiro$imdb_score)
max(ficheiro$imdb_score)

write.csv(pergunta4,"ItemList.csv", row.names = TRUE)

txn4 = read.transactions(file="ItemList.csv", rm.duplicates= TRUE,sep=",");

summary(txn4)
inspect(txn4[1:10])

rules.4 <-  apriori(txn4,parameter = list(support =0.0005,
confidence = 0.60,target = "rules", minlen = 2))

df.4 <- as(rules.4, "data.frame")
df.4[order(df.4$lift, df.4$confidence), ]

summary(rules.4)

inspect(sort(rules.4 , by = "lift")[1:15])
inspect(rules.4)
inspect(rules.4[1:15])
```

G Código R Modelo 5

```
library(grid)
library(gdata)
library(arulesViz)
library(Matrix)
library(arules)
set.seed(123456)

ficheiro<-read.csv("outMovieFinal.csv",header = T, stringsAsFactors = TRUE)

for(i in 29:length(ficheiro)) {

  ficheiro[is.na(ficheiro[i]),i]<-0
}

ficheiro<-ficheiro[,c(-4,-16,-17,-18,-27,-10)]
array<-which(ficheiro[2]=="", arr.ind = T)
ficheiro<-ficheiro[-array[,1],]
array<-which(ficheiro[1]=="", arr.ind = T)

for(i in 1:length(array[,"row"])){
  ficheiro[array[i,1],"color"]<-"Color"
}

for(i in 1:length(ficheiro)){
  array<-which(is.na(ficheiro[i]),arr.ind = T)
  if(length(array)>0) {
    ficheiro<-ficheiro[-array[,1],]
  }
}
cenas2=c(0)

pergunta5<-ficheiro[,c(2,9,8,18)]
pergunta5[,"Lucro"]<-cenas2

pergunta5$Lucro<-pergunta5$gross - pergunta5$budget
par(mfrow=c(1,2))

barplot(table(pergunta5$Lucro))

ARRAY<- which(pergunta5[5]>0,arr.ind = T)

pergunta5=pergunta5[,-c(3,4)]

for(i in 1:nrow(pergunta5)){
  if(pergunta5[i,"Lucro"]<0) {
    pergunta5[i,"Lucro"]<-"Prejuizo"
  }

  if(pergunta5[i,"Lucro"]==0) {
    pergunta5[i,"Lucro"]<-"Lucro zero"
```

```
  }
}

pergunta5[ARRAY[,"row"],"Lucro"]<-"Lucro"

barplot(table(pergunta5$Lucro))

write.csv(pergunta5,"Idade.csv", row.names = TRUE)

txn = read.transactions(file="Idade.csv", rm.duplicates= TRUE,sep=",")

summary(txn)
inspect(txn[1:10])

rules <-  apriori(txn,parameter = list(support =0.0005,
confidence = 0.6,target = "rules", minlen = 2))

df <- as(rules, "data.frame")
df[order(df$lift, df$confidence), ]

summary(rules)

inspect(sort(rules , by = "lift")[1:15])

inspect(rules)
inspect(rules[1:10])
```