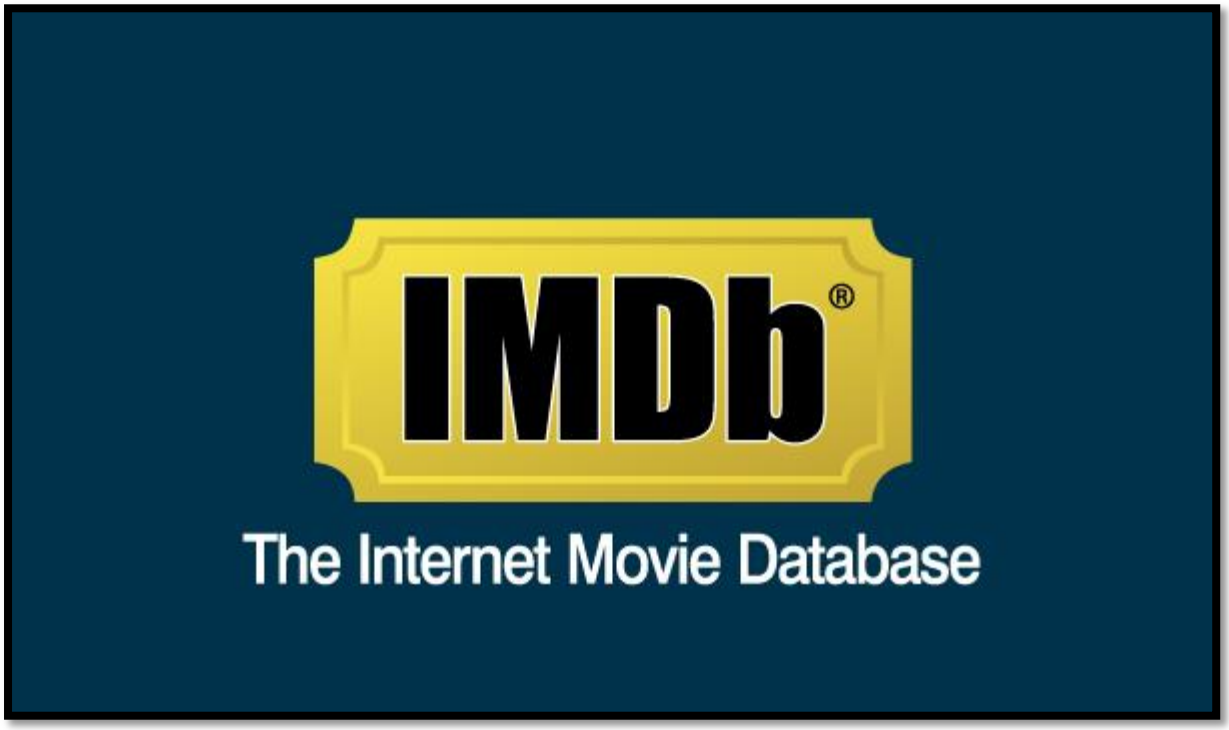
Relatório Projeto Final- Ciência dos dados



André Tavernaro
Daniel Zelv Freund
Guilherme Batista

Sumário

Introdução_____3

Objetivos_____3

Ferramentas utilizadas _____3

Metodologia_____6

Resultados_____11

Conclusão e Aplicações_____12

Referências_____13

Introdução

Com o surgimento da indústria cinematográfica e seu desenvolvimento acelerado ao longo dos anos, diversas plataformas e produtoras de filmes surgiram para revolucionar esse mercado, produzindo filmes em massa para atender a atual população internacionalizada do mundo atual. A ideia do projeto surgiu justamente desse fenômeno de produção em massa de filmes, em que os filmes recebem notas específicas conforme o sucesso que cada um atinge.

Objetivos

A fim de desenvolver as habilidades críticas, destreza no manuseamento de altos volumes de informação e trazer uma maior familiaridade do aluno com o processo que um cientista de dados passa em seu cotidiano. O projeto trouxe a oportunidade de se utilizarem regressões, classificadores ou clusters para fazer previsões sobre novas informações.
O tema escolhido foi a regressão linear, usada para prever as notas do IMDB (um dos mais relevantes sites de avaliações cinematográficas mundial). Devido ao uso de diversas variáveis em sua análise, trata-se de um modelo de *Regressão Múltipla*.

Ferramentas Utilizadas

- **Regressão Linear Múltipla**

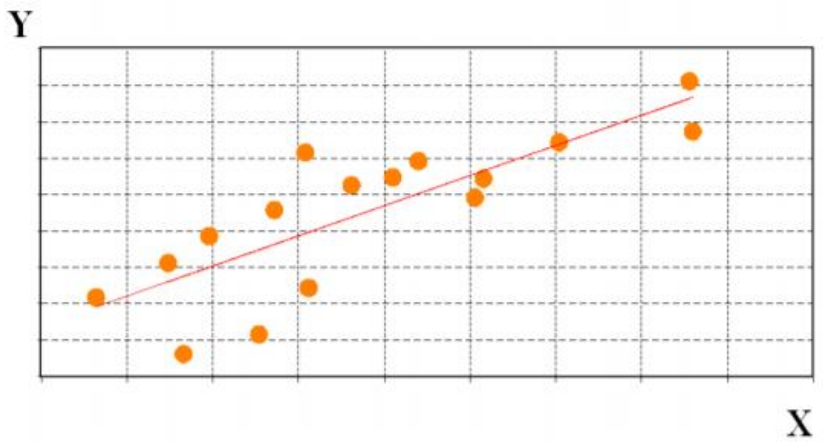
A regressão linear múltipla funciona da seguinte forma: primeiro escolhemos uma variável que queremos prever, depois, coletamos variáveis que possuem forte relação com a escolhida (com comportamento linear), e assim, conseguimos prever de certa forma a variável escolhida para outros dados.

Genericamente, um modelo de regressão linear múltipla com k variáveis independentes e p parâmetros (p=k+1) pode ser representado por:

$$Y_i = \alpha + \beta_1 X_{1i} + \beta_2 X_{2i} + \ldots + \beta_k X_{ki} + e_i$$

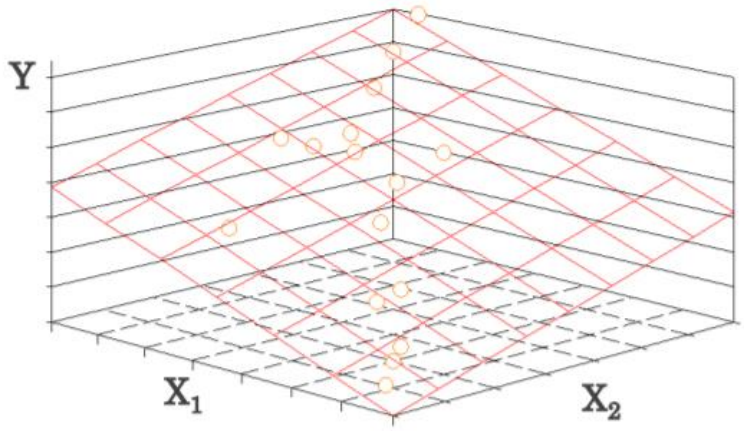
- α é o valor esperado de Y quando todos as variáveis independentes forem nulas;
- β_1 é a variação esperada de Y dado um incremento unitário em X_1 , mantendo constantes todas as demais variáveis independentes;
- β_k é a variação esperada de Y dado um incremento unitário em X_k , mantendo constantes todas as demais variáveis independentes;
- e_i é o erro não explicado pelo modelo;

Gráfico de uma **regressão linear simples:**



Note que o gráfico é **bidimensional**. O modelo tenta ajustar uma reta que melhor se adeque ao comportamento dos pontos, tentando reduzir ao máximo os resíduos.

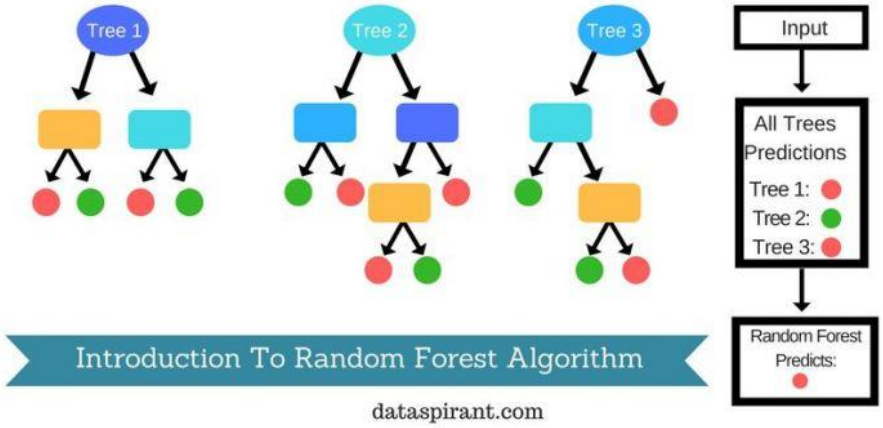
Gráfico de uma **regressão linear múltipla:**



Neste caso o modelo de regressão linear múltipla aborda três variáveis, logo o gráfico possui um plano para cada variável. Tendo sua regressão representada pelo plano entre as três variáveis (X1, X2, Y).

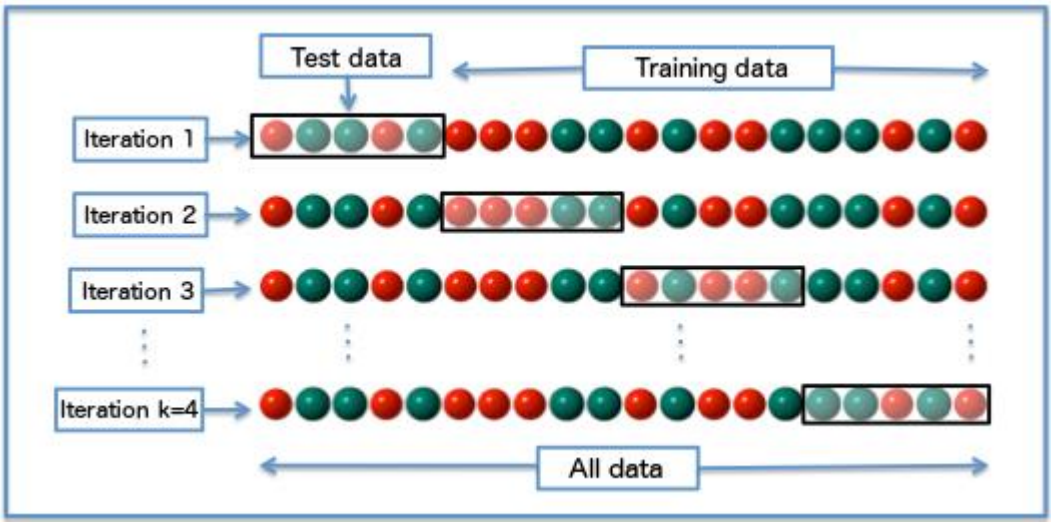
- **Random Forest**

Usado tanto para problemas de regressão como classificadores, o Random Forest cria a partir de um conjunto de dados de treinamento regras de tomada de decisão e verifica se onde os valores de teste mais se adequeam dentro deles.



- ❑ **Cross Validation**

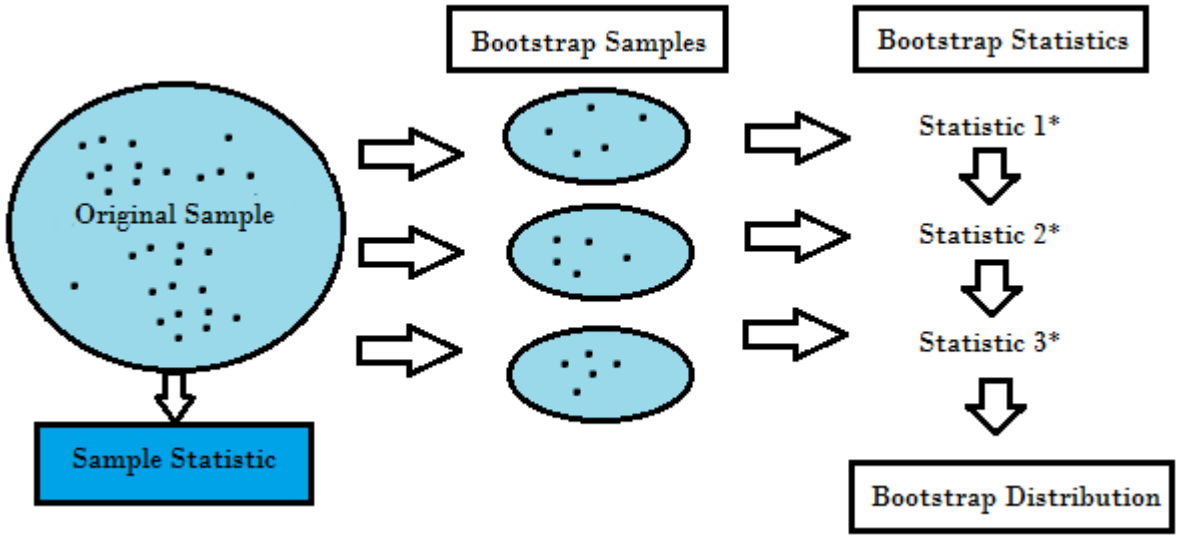
Este método divide nosso dataset em várias partições, realizando um teste e um treinamento para cada partição e, obtendo uma precisão para cada uma. Deste modo, fizemos dez partições e calculamos a média das precisões das dez partições, obtendo assim, uma precisão maior para nosso modelo.



- ❑ **Bootstrapping**

Bootstrapping é uma tecnica estatistica que usa de um rearranjo dos dados e repetição entre valores de uma amostra de um conjunto amostral a fim de fazer

estimativas sobre o conjunto inteiro, possuindo um intervalo de confiança para a previsão.



Metodologia

- ❏ **Limpeza dos dados**

Para iniciar a limpeza dos dados, primeiro foram removidos os dados duplicados e escolhidas as colunas relevantes para nossa análise.
["title","runtime","budget","revenue","vote_average","vote_count","popularity" ,"genres",'release_date']

Depois, por se tratar de uma regressão, **genres** e **release_date** foram convertidos para valores numéricos:

A coluna **genres** contém um dicionário que informa os gêneros em que o filme se encaixa. O problema disso é que os dados não são numéricos. Desse modo foi criado uma coluna para cada gênero, com 0 sendo quando um filme não se encaixa na categoria e 1 quando se encaixa.

A coluna **release_date** contém a data em que o filme foi lançado, porém ele se encontra no formato “DD-MM-YY, fazendo com que o compilador a interprete como uma string. Logo ela foi particionada em três colunas : Ano, Mês, Dia (integers).

- ❏ **Limpeza de valores nulos (0) e inválidos (NA)**

Ao se analisar os valores nulos em cada uma das colunas selecionadas, notou-se uma relevante quantidade de valores nulos:

```
verificador_nulo(df_analise)
```

```
35 runtime
1037 budget
1427 revenue
63 vote_average
62 vote_count
1 popularity
```

Devido à alta quantidade de valores nulos de *budget* e *revenue*, foi-se optado por **remover** seus filmes do dataframe.
Ao se removerem os valores restantes de *vote_average* e *vote_count*, notou-se uma **queda** no r^2 de nosso modelo, e ao substituí-los pela mediana de suas colunas, r^2 **não** se alterou, portanto elas foram **intocadas** no restante do projeto.

- ❏ **Variáveis do dataframe (formatada)**

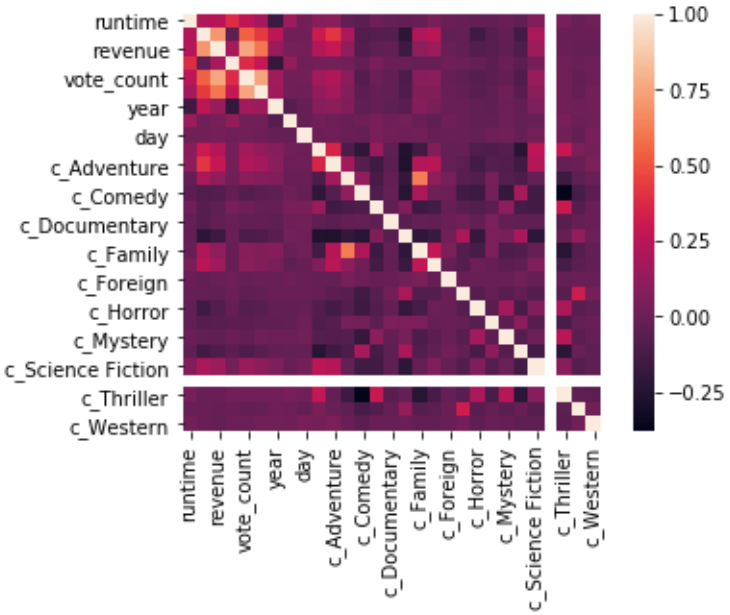
Title	**Título do filme**
Runtime (minutos) Contagem total: 3164 desvio padrão: 20,67 média: 110,26 mínimo: 41 máximo: 338	**Tempo de filme**
Budget (dólares) Contagem total: 3164 desvio padrão:3.666 e+07 média: 3.706 e+07 mínimo: 3.666 e+07 máximo: 1.750 e+08	**Orçamento**
Revenue (dólares)	**Receita**

Contagem total: 3164 média: 1.083 e+08 desvio padrão:1.520 e+08 mínimo: nan máximo: 1.513 e+08	
Vote_average (valor absoluto, média do número de votos) Contagem total: 3164 média: 6,30 desvio padrão: 0,876 mínimo: 0,00 máximo: 8.5	**Média dos votos**
Vote Count (valor absoluto, número de votos) Contagem total: 3164 média: 903,08 desvio padrão: 1280,86 mínimo: 0,00 máximo: 13752	**Número de votos das pessoas**
Popularity (índice que varia de 0 a 100) Contagem total: 3164 média: 27,84 desvio padrão: 35,10 mínimo: 0,02 máximo: 87,5	**Popularidade**
Genres (dicionário com as categorias em que o filme se enquadra)	**Gêneros de filme**
C_Gênero_x (categorias do filme divididas)	**0 pertence** **1 não pertence**
Ano (1916-2016) Contagem total: 3164	**Ano em que o filme foi lançado**

média: 2001,49 desvio padrão: 13,33 mínimo: 1916 máximo: 2016	
<u>Mês</u> (1-12) Contagem total: 3164 média: 7 desvio padrão: 3,37 mínimo: 1 máximo: 12	**Mês em que o filme foi lançado**
<u>Dia</u> (1-31) Contagem total: 3164 média: 15,49 desvio padrão: 8,47 mínimo: 1 máximo: 31	**<u>Dia</u> em que o filme foi lançado**

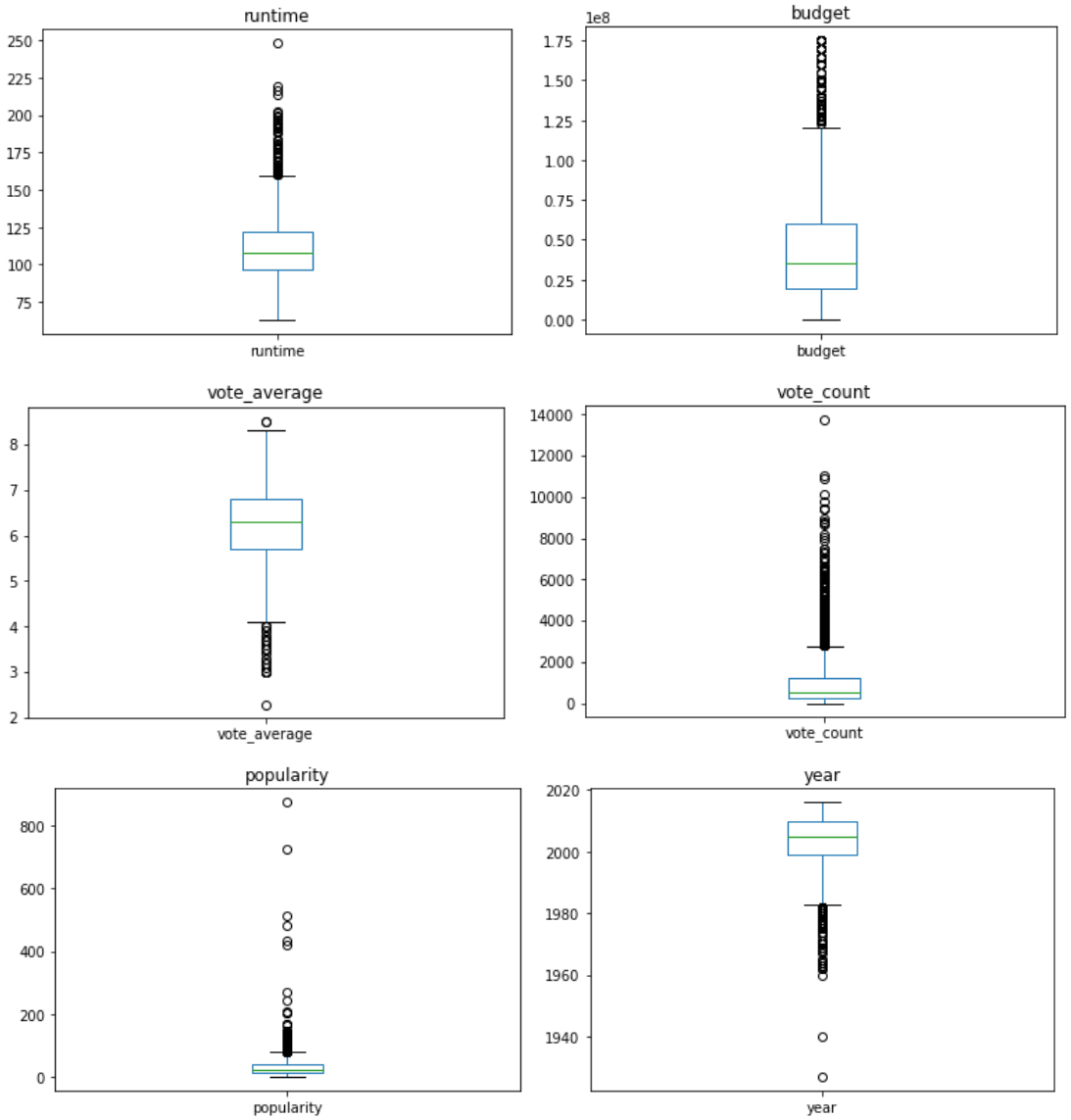
- **Análise gráfica**

  - **Heatmap**



Com a ajuda do gráfico *heatmap*, é possível notar o padrão de regressão nas variáveis observadas, permitindo a continuação da análise exploratória.

  - **Boxplot**



Com os **boxplots**, é possível se notar a alta quantidade de outliers em todas as variáveis de nosso modelo.
Em algumas delas, como *revenue* e budget, por se tratarem de variáveis com valores muito altos, foram testados modelos logarítmicos para tentar aumentar a precisão do modelo, mas como isso não ocorreu, foram mantidas **inalteradas**. Ao mesmo tempo, ao se removerem **98%** dos valores mais altos de budget, obteve-se um aumento substancial em r^2**.**

title	runtime	budget	revenue	vote_average	vote_count	popularity	year	month	day
The Good Dinosaur	93.0	175000000	331926147	6.6	1736	51.692953	2015.0	11.0	14.0
Up	96.0	175000000	735099082	7.7	6870	92.201962	2009.0	5.0	13.0
Monsters vs Aliens	94.0	175000000	381509870	6.0	1423	36.167578	2009.0	3.0	19.0
Suicide Squad	123.0	175000000	745000000	5.9	7458	90.237920	2016.0	8.0	2.0
Evan Almighty	96.0	175000000	173000000	5.3	1151	27.082182	2007.0	6.0	9.0
Waterworld	135.0	175000000	264218220	5.9	992	44.640292	1995.0	7.0	28.0
G.I. Joe: The Rise of Cobra	118.0	175000000	302469017	5.6	1962	32.852443	2009.0	8.0	4.0
Inside Out	94.0	175000000	857611174	8.0	6560	128.655964	2015.0	6.0	9.0
The Jungle Book	106.0	175000000	966550600	6.7	2892	94.199316	2016.0	4.0	7.0
47 Ronin	119.0	175000000	150962475	5.9	1326	41.796339	2013.0	12.0	6.0

❑ Describe

	runtime	budget	revenue	vote_average	vote_count	popularity	year	month	day
count	2506.000000	2.506000e+03	2.506000e+03	2506.000000	2506.000000	2506.000000	2506.000000	2506.000000	2506.000000
mean	111.569034	4.574538e+07	1.298196e+08	6.246848	1024.930966	30.722264	2003.577813	7.041101	15.458500
std	20.071793	3.650087e+07	1.624954e+08	0.846625	1365.601579	37.731164	9.403981	3.394181	8.404784
min	63.000000	1.000000e+01	1.100000e+01	2.300000	2.000000	0.034259	1927.000000	1.000000	1.000000
25%	97.000000	1.902500e+07	2.923493e+07	5.700000	225.000000	12.450194	1999.000000	4.000000	9.000000
50%	108.000000	3.500000e+07	7.321280e+07	6.300000	537.500000	22.778388	2005.000000	7.000000	15.000000
75%	122.000000	6.000000e+07	1.629186e+08	6.800000	1252.000000	38.816247	2010.000000	10.000000	22.000000
max	248.000000	1.750000e+08	1.513529e+09	8.500000	13752.000000	875.581305	2016.000000	12.000000	31.000000

❑ Separação das variáveis

As variáveis de nosso modelo foram separadas em:

1. **Variável dependente (Y)** – Variável da previsão (vote_average).

2. **Variável independente (X**) – Variáveis relacionadas a varíavel dependente. (nota do imdb)

❑ Regress

Com o uso da função *regress e summary*, e adotando um *alpha* de **5%**, as variáveis que não obtiveram seu P>|t| mínimo foram descartadas. (*c_genero* e *day*)

Dep. Variable:	vote_average	R-squared:	0.403
Model:	OLS	Adj. R-squared:	0.398
Method:	Least Squares	F-statistic:	79.87
Date:	Sun, 18 Nov 2018	Prob (F-statistic):	1.90e-259
Time:	15:50:41	Log-Likelihood:	-2492.4
No. Observations:	2507	AIC:	5029.
Df Residuals:	2485	BIC:	5157.
Df Model:	21		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	18.6573	3.025	6.168	0.000	12.726	24.589
runtime	0.0139	0.001	20.183	0.000	0.013	0.015
budget	-6.833e-09	5e-10	-13.654	0.000	-7.81e-09	-5.85e-09
revenue	1.322e-10	1.36e-10	0.974	0.330	-1.34e-10	3.98e-10
vote_count	0.0003	1.68e-05	17.179	0.000	0.000	0.000
popularity	0.0006	0.001	1.202	0.230	-0.000	0.002
c_Action	0.0418	0.035	1.209	0.227	-0.026	0.110
c_Adventure	0.0188	0.037	0.502	0.616	-0.055	0.092
c_Animation	0.1081	0.059	1.842	0.066	-0.007	0.223
c_Comedy	0.0203	0.031	0.645	0.519	-0.041	0.082
c_History	0.0937	0.068	1.371	0.171	-0.040	0.228
c_Horror	-0.0562	0.048	-1.180	0.238	-0.150	0.037
c_Music	-0.1572	0.076	-2.067	0.039	-0.306	-0.008
c_Mystery	0.0241	0.050	0.486	0.627	-0.073	0.121
c_Romance	-0.0638	0.038	-1.688	0.092	-0.138	0.010
c_Science Fiction	0.0031	0.041	0.075	0.940	-0.077	0.083
c_TV Movie	1.239e-16	2.27e-17	5.465	0.000	7.94e-17	1.68e-16
c_Thriller	-0.0186	0.034	-0.542	0.588	-0.086	0.049
c_War	0.0744	0.073	1.013	0.311	-0.070	0.218
c_Western	-0.1104	0.110	-1.003	0.316	-0.326	0.106
month	0.0225	0.004	5.724	0.000	0.015	0.030
year	-0.0070	0.002	-4.694	0.000	-0.010	-0.004
day	-0.0020	0.002	-1.246	0.213	-0.005	0.001

Omnibus:	235.306	Durbin-Watson:	2.001
Prob(Omnibus):	0.000	Jarque-Bera (JB):	410.043
Skew:	-0.654	Prob(JB):	9.13e-90
Kurtosis:	4.488	Cond. No.	8.65e+25

- **Divisão TESTE | TREINAMENTO**

Com as variáveis limpas e selecionadas, os filmes foram separados em teste e treinamento (1/3 teste | 2/3 treinamentos). Fazendo com que o algoritmo não se auto alimente, distorcendo os resultados.

Resultados

r^2:

- **Método Regressão Linear: 0.43**
- **Método Random Forest: 0.49**

Cross Validation (r^2):

- **Método Regressão Linear: 0.37**
- **Método Random Forest: 0.42**

Bootstrapping no Cross Validation(r^2):

- ❏ **Método Regressão Linear: 0.37**
- ❏ **Método Random Forest: 0.42**

Intervalo de confiança do Bootstrapping no Cross Validation(r^2) (95%):

- ❏ **Método Regressão Linear: [0.36, 0.38]**
- ❏ **Método Random Forest: [0.41, 0.43]**

Conclusão e Aplicações

Depois dos dados serem limpos e analisados, foram aplicados dois métodos: *OLS (regressão), Random Forest*. Devido ao seu melhor desempenho*, Random Forest* foi o escolhido como primário para nosso algoritmo. Entretanto sua precisão foi de apenas **49%**. Tendo como principal suspeita a **falta de dados** causados por valores nulos ou inválidos, que acabou limitando a base de dados de **5000 filmes.** Como as variáveis possuíam valores muito diferentes, seriam necessárias quantidades mais altas de dados para análise. Outro fator que deve ser levado em consideração foram as datas dos filmes que se iniciavam desde 1916, que possuíam bilheterias, receitas e orçamentos diferentes dos atuais. Entretanto devido a quantidade de dados, a remoção destes restringiria muito o campo de análise.
Também foram utilizados do *cross validation* para verificar a **eficácia** dos modelos, e *bootstrapping* (95% de intervalo de confiança) para fazer previsões mais precisas sobre o cross validation. Obtendo o resultado melhor com o *RandomForest (r^2 com bootstrapping do cross validation de [0.41, 0.43]).*
Devido a sua alta aplicabilidade por parte tanto de cineastas para checar sua possível nota em um site de alta credibilidade como o IMDB, como de fãs para ver como seria o próximo filme de sua saga favorita, seria possível fazer uma futura iteração com uma arrecadação de fundos, a fim de obter mais dados para que a análise se tornasse mais precisa. Já que o site disponibiliza um datasheet com mais de 200k de filmes, porém possui um custo muito alto para sua utilização.

Referências

http://www.portalaction.com.br/analise-de-regressao/regressao-linear-multipla
https://www.surveyanalytics.com/system/heatmap.html
https://www.kaggle.com/adktyakirloskar/movies

http://dataaspirant.com/2017/05/22/random-forest-algorithm-machine-learing/
https://scikit-learn.org/stable/modules/cross_validation.html