PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA
Pós-graduação *Lato Sensu* em Ciência de Dados e Big Data

Bárbara Gonçalves Oliveira

FRESCO OU PODRE: UMA ANÁLISE DOS *REVIEWS* DO *ROTTEN TOMATOES*

Belo Horizonte
2022

Bárbara Gonçalves Oliveira

FRESCO OU PODRE: UMA ANÁLISE DOS *REVIEWS* DO *ROTTEN TOMATOES*

Trabalho de Conclusão de Curso apresentado ao Curso de Especialização em Ciência de Dados e Big Data como requisito parcial à obtenção do título de especialista.

Belo Horizonte

2022

SUMÁRIO

1. Introdução

1.1. Contextualização

Desde que consigo me recordar, filmes fazem parte da minha vida. Considero-me uma fã de filmes densos, difíceis e que dividem a opinião pública. Nesse contexto, sempre que converso com outras pessoas sobre filmes polêmicos, o resultado não poderia ser diferente da classificação desses. Um consenso sempre esteve longe de ser atingido!

Com isso em mente, veio-me a ideia de aliar ciência de dados a esse problema a fim de tentar desvendar essas polêmicas por meio de dados, tentar achar um consenso e descobrir se alguns filmes são amados por todos ou só por alguns. Para isso, decidi criar um modelo para aliar opiniões de críticos certificados a opiniões de telespectadores em geral.

Os dados de críticos certificados utilizados são provenientes do *Rotten Tomatoes*, um site que de acordo com ele mesmo, é, junto com o *Tomatometer,*

> "a fonte mais confiável do mundo de recomendações de entretenimento de qualidade. Como o principal agregador online de resenhas de críticos de filmes e programas de TV, oferecem aos fãs um guia completo do que é fresco – e do que é podre – nos cinemas e em casa."[1] (Rotten Tomatoes, 2021)

A base do *Rotten Tomatoes* foi coletada da plataforma *Kaggle* e é composta por dois conjuntos de dados: dados relacionados aos filmes e dados relacionados aos *reviews* dos críticos. Esses dados foram raspados da web no final de 2020 e estão disponibilizados na plataforma Kaggle para fins de análise.

Nessa base de dados, como dito acima, além dos *reviews* dos críticos, há também a classificação ou rótulo dada por estes aos filmes, que pode ser *Rotten* quando o filme é ruim ou *Fresh* quando o filme é bom.

Os dados de revisão dos telespectadores em geral foram retirados do site do *Rotten Tomatoes*, através de uma raspagem feita para cada filme a ser analisado.

[1] Traduzido do original: "Rotten Tomatoes and the Tomatometer score are the world's most trusted recommendation resources for quality entertainment. As the leading online aggregator of movie and TV show reviews from critics, we provide fans with a comprehensive guide to what's Fresh – and what's Rotten – in theaters and at home.". Disponível em: https://www.rottentomatoes.com/about, acesso em 26/06/2021.

Além das revisões de críticos, este site também possibilita usuários comuns - telespectadores em geral, a deixar suas próprias opiniões sobre os filmes.

1.2. O problema proposto

O problema a ser resolvido através dos dados aqui, que foi brevemente descrito na contextualização, é tentar responder opiniões polêmicas sobre filmes igualmente polêmicos e chegar a um consenso que alie opiniões de críticos, que podem ser tomados aqui como pessoas com maior entendimento de cinema e artes, e opiniões de telespectadores em geral, que podem enriquecer essa análise e democratizá-la um pouco mais.

Seguindo a linha dos 5-Ws, abaixo são respondidas as perguntas propostas:

- Por que coletar, analisar e comparar informações de filmes?

Toda análise surge por conta de uma dúvida, uma questão a ser resolvida. Nesse caso, essa análise é importante porque ajuda a resolver polêmicas sobre se um filme polêmico é considerado bom ou não, num contexto geral - e não somente entre críticos. Além disso, é um tópico de meu interesse que se alia ao conhecimento obtido durante as disciplinas desse curso de pós-graduação.

- Quem/quais são esses dados?

Os dados coletados vieram de diversas fontes, porém todos estavam disponibilizados online para esses fins:

1. *Rotten Tomatoes Movies and Critic Review Dataset*: Esse conjunto de dados foi obtido através da plataforma *Kaggle*. Os dados foram previamente raspados do website do *Rotten Tomatoes.* Esses dados são públicos e foram raspados em 31 de outubro de 2020. Possui um conjunto de dados com informações dos filmes em geral e um conjunto de dados com as informações dos reviews, incluindo o comentário do crítico, e o status do review, *Fresh* e *Rotten*.

2. *Rotten Tomatoes* - *Reviews* dos Usuários*:* Esses dados foram obtidos através de uma raspagem do próprio website, feita manualmente para cada filme a ser levado em consideração na análise final.

- Qual o objetivo dessa análise?

  O objetivo dessa análise é extrair o texto e encontrar padrões nos comentários dos críticos que possam indicar corretamente os rótulos *Fresh* e *Rotten* dados pelos críticos. Utilizando de métodos supervisionados de aprendizado de máquina para obter um modelo treinado que posteriormente poderá ser utilizado para interpretar outros comentários e classificá-los entre *Fresh* e *Rotten*.

- Quais os aspectos geográficos dessa análise?

  A base utilizada para treino, apesar de conter diversos filmes de diversos países, é americana e todo o conteúdo está em inglês, portanto, é esperado que se tenha uma visão mais ocidental nas análises.

- Qual o período que está sendo analisado?

  Os comentários analisados são dos últimos 20 anos, mais precisamente do ano 2000 até o ano de 2020, quando esses dados foram raspados. Já os filmes revisados aqui, são de um período muito mais amplo – de 1914 até 2020.

2. Coleta de Dados

Os dados utilizados neste trabalho são provenientes de duas fontes distintas, abaixo são detalhados os métodos de coleta e as variáveis presentes em cada um deles.

2.1. *Rotten Tomatoes Movies and Reviews Dataset*

A base de dados do *Rotten Tomatoes* está disponibilizada em .csv no site do *Kaggle*[2] e foi obtida em 26/06/2021. Contém dois conjuntos de dados: *movies* e *critic_reviews*.

1. Conjunto de dados *Movies*

Esse conjunto possui dados de filmes com data de lançamento de 1914 a 2020, é composto previamente por 17.712 registros e 22 colunas.

Nome da coluna/campo	**Descrição**	**Tipo**
Rotten_tomatoes_link	URL do filme no Rotten Tomatoes website.	Object
Movie_title	Título do filme em inglês.	Object
Movie_info	Breve sinopse do filme.	Object
Critics_consensus	Consenso da avaliação dos críticos.	Object
Content_rating	Classificação indicativa de idade do filme.	Object
Genres	Gêneros do filme.	Object
Directors	Diretores do filme.	Object
Authors	Autores do filme.	Object
Actors	Atores que atuaram no filme.	Object
Original_release_date	Data original de estreia do filme nos Estados Unidos.	Object

[2] https://www.kaggle.com/stefanoleone992/rotten-tomatoes-movies-and-critic-reviews-dataset

Streaming_release_date	Data de início do streaming do filme.	Object
Runtime	Duração do filme em minutos.	Object
Production_company	Produtora do filme.	Object
Tomatometer_status	Avaliação qualitativa do filme entre *Rotten* e *Fresh.*	Object
Tomatometer_rating	Percentual de avaliações positivas do filme.	Float64
Tomatometer_count	Número de críticas totais utilizadas para o cálculo da variável tomatometer_status.	Float64
Audience_status	Classificação qualitativa do filme pelos telespectadores usuários do website.	Object.
Audience_rating	Percentual de avaliações positivas dos usuários.	Float64
Audience_count	Número de avaliações totais dos usuários utilizadas para o cálculo da variável audience_status.	Float64
Tomatometer_top_critics_count	Número de avaliações de top críticos.	Int64
Tomatometer_fresh_critics_count	Número de avaliações do tipo *Fresh.*	Int64
Tomatometer_rotten_critics_count	Número de avaliações do tipo *Rotten.*	Int64

2. Conjunto de dados *Critic_reviews*

Esse conjunto possui dados das revisões dos filmes disponíveis no *Rotten Tomatoes,* os mesmos filmes do conjunto de dados *Movies*. É composto previamente por 1.130.017 registros e 8 colunas.

Nome da coluna/campo	**Descrição**	**Tipo**
Rotten_tomatoes_link	URL do filme no Rotten Tomatoes website.	Object
Critic_name	Nome do crítico que escreveu a revisão.	Object
Top_critic	Valor booleano que indica se esse crítico é um top crítico ou não.	Bool
Publisher_name	Nome da publicadora na qual o crítico trabalha.	Object
Review_type	Classificação do filme de acordo com o crítico, entre *Fresh* e *Rotten.*	Object
Review_score	Nota avaliativa dada pelo crítico ao filme. Não possui uma consistência.	Object
Review_date	Data da revisão.	Object
Review_content	Conteúdo da revisão em si.	Object.

2.2. *Rotten Tomatoes* - Reviews dos Usuários

Para coletar esses dados, foi necessário utilizar uma chamada na API do próprio site do *Rotten Tomatoes*, visto que esses dados não foram encontrados já disponibilizados na web. Essa chamada foi feita pela última vez em 28 de abril de 2022.

A raspagem de dados feita foi direcionada para cada filme que fosse ter os *reviews* de usuários classificados pelo modelo. O filme utilizado neste trabalho foi o filme “Mãe!”. Como resultado da coleta, foram obtidas 3.678 entradas de *reviews*.

A estrutura da tabela seguiu o padrão já imposto pelos conjuntos de dados obtidos da plataforma *Kaggle*.

Nome da coluna/campo	**Descrição**	**Tipo**
Rotten_tomatoes_link	URL do filme no Rotten Tomatoes website.	Object
Review_content	Conteúdo da revisão em si.	Object

3. Processamento/Tratamento de Dados

1. Conjunto de dados *Critic_reviews*

Para treinamento do modelo foram utilizados dados do *dataset* de *reviews* apenas. As colunas utilizadas foram *review_content* e *review_type* que, correspondem, respectivamente ao texto da avaliação do crítico e ao tipo de classificação dada por este ao filme entre *Fresh* e *Rotten.* Esse dataset, composto por 1.130.017 registros possuía 65.806 textos de *reviews* nulos, que por sua vez, foram removidos dos dados. Em relação ao rótulo dos dados, o dataset se apresenta levemente desbalanceado, com 64% dos rótulos sendo *Fresh* e os outros 36% sendo *Rotten*, conforme figura 1.

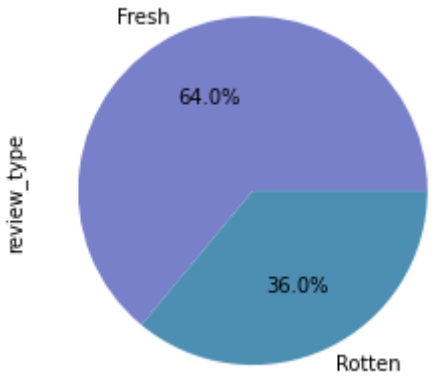


Figura 1: Distribuição dos dados de rótulo

A coluna *review_score* não foi utilizada por apresentar dados de classificações que utilizaram parâmetros diferentes. Conforme exemplificado na figura 2.

```
In [75]: rt_reviews.review_score.value_counts()
Out[75]: 3/5          76972
         4/5          73702
         3/4          69641
         2/5          50130
         2/4          45359
                      ...
         3.65/10          1
         1.5/6            1
         9.2021/10        1
         8.56/10          1
         5.76/10          1
         Name: review_score, Length: 813, dtype: int64
```

Figura 2: parâmetros diferenciados utilizados na variável review_score

Foi criada uma coluna para fins de análise exploratória, chamada *review_year,* baseada na data do review (*review_date*).

O restante dos dados do *dataset* de *review* não foi utilizado por não ser relevante à modelagem.

2. Conjunto de dados *Movies*

O conteúdo do conjunto de dados *Movies* foi utilizado apenas como um suporte para entendimento dos dados, visto que nele temos informações detalhadas acerca dos filmes, como por exemplo gênero, atores, diretores, produtoras, data de estreia, etc.

Como a maioria dessas informações já estava bem categorizada, poucos processamentos foram necessários. A separação dos tipos de gêneros que cada filme se enquadra foi um deles. Por exemplo, o filme *Mother!,* como visto na figura 3, é elencado como Drama, Horror, Mistério & Suspense. Para fins de facilitar a análise exploratória, foi feito o tratamento desse campo de modo a manter apenas o primeiro gênero.

	rotten_tomatoes_link	movie_title	movie_info	critics_consensus	content_rating	genres	directors
10619	m/mother_2017	mother!	A young woman spends her days renovating the V...	There's no denying that mother! is the thought...	R	Drama, Horror, Mystery & Suspense	Darren Aronofsky

Figura 3: Registro do filme Mother! e sua classificação de acordo com os gêneros.

Abaixo, na figura 4, pode-se ver como esse *dataset* contém registros com múltiplos gêneros.

```
rt_movies.genres.value_counts()
```

```
Drama                                                                    1887
Comedy                                                                   1263
Comedy, Drama                                                             863
Drama, Mystery & Suspense                                                 731
Art House & International, Drama                                          589
                                                                         ...
Art House & International, Classics, Cult Movies, Horror, Science Fiction & Fantasy    1
Action & Adventure, Cult Movies, Drama, Science Fiction & Fantasy         1
Art House & International, Documentary, Sports & Fitness                  1
Action & Adventure, Drama, Mystery & Suspense, Special Interest           1
Action & Adventure, Drama, Horror, Kids & Family, Mystery & Suspense      1
Name: genres, Length: 1106, dtype: int64
```

Figura 4: Classificação de gênero dos filmes mostrando múltiplos gêneros por filme.

Para solucionar isto, foi feito o split dos dados, conforme mostra a figura 5, mantendo apenas o primeiro registro, deste modo, pôde-se obter um melhor aproveitamento dos gêneros para se observar em um gráfico em etapas posteriores.

```
rt_movies['split_genres'] = rt_movies.genres.str.split(',').str[0]
```

```
rt_movies.split_genres.value_counts()
```

```
Drama                       3789
Comedy                      3725
Action & Adventure          3551
Art House & International    2021
Documentary                 1725
Classics                    1110
Horror                       943
Animation                    379
Mystery & Suspense           289
Kids & Family                 46
Science Fiction & Fantasy     40
Musical & Performing Arts     26
Cult Movies                   22
Romance                       14
Western                        9
Special Interest               3
Television                     1
Name: split_genres, dtype: int64
```

Figura 5: Tratamento dos dados utilizando a função "split" e resultado do tratamento.

A coluna *original_release_date* tem as datas de lançamento dos filmes, um dado interessante de ser observado, Porém, há registros nulos e por isso foi necessário fazer uma limpeza desses dados. Ao invés de excluí-las e para não impactar na análise dos outros campos, essas datas nulas foram preenchidas com uma data fictícia que pode ser facilmente ignorada em etapas posteriores. Para facilitar o entendimento, um novo campo chamado *movie_year* foi criado para coletar apenas o ano de lançamento de cada registro.

```
[21] rt_movies.original_release_date.fillna('2222-02-02', inplace=True)
```

```
[22] rt_movies['original_release_date'] = pd.to_datetime(rt_movies['original_release_date'])
```

```
[23] max(rt_movies.original_release_date.loc[rt_movies.original_release_date != '2222-02-02'])

     Timestamp('2020-09-30 00:00:00')
```

```
[24] min(rt_movies.original_release_date)

     Timestamp('1914-06-01 00:00:00')
```

```
[25] rt_movies['movie_year'] = rt_movies.loc[rt_movies.original_release_date != '2222-02-02',
                                             'original_release_date'].apply(lambda x: x.year)
```

4. Análise e Exploração dos Dados

Em primeira análise, foi feita a verificação de que a quantidade de filmes batia nos dois *datasets*. O conjunto de dados *review* é extensamente maior do que o conjunto de dados *movies*. Isso se dá pois no primeiro temos várias revisões por filmes, enquanto que no segundo temos apenas uma linha por filme.

```
rt_reviews.groupby(["rotten_tomatoes_link"])["review_content"].count().sort_values(ascending=False).head(10)
```

```
rotten_tomatoes_link
m/star_wars_the_rise_of_skywalker           992
m/solo_a_star_wars_story                    948
m/star_wars_the_last_jedi                   946
m/rogue_one_a_star_wars_story               892
m/spider_man_far_from_home                  880
m/star_wars_episode_vii_the_force_awakens   874
m/ready_player_one                          866
m/shazam                                    806
m/spider_man_homecoming                     780
m/roma_2018                                 774
Name: review_content, dtype: int64
```

Figura 6: Agrupamento de filmes no dataset de reviews por número de reviews.

Em relação aos gêneros, foi feito um cruzamento com a classificação final do filme, o que pode ser visto na figura 7. Nota-se o maior volume de filmes sendo de Ação & Aventura, seguidos de Drama e depois Comédia. A maioria dos gêneros tem uma proporção maior de filmes considerados *Fresh*. O gênero Horror, particularmente, parece ser o que mais divide opiniões, mostrando que praticamente metade dos reviews são Rotten.

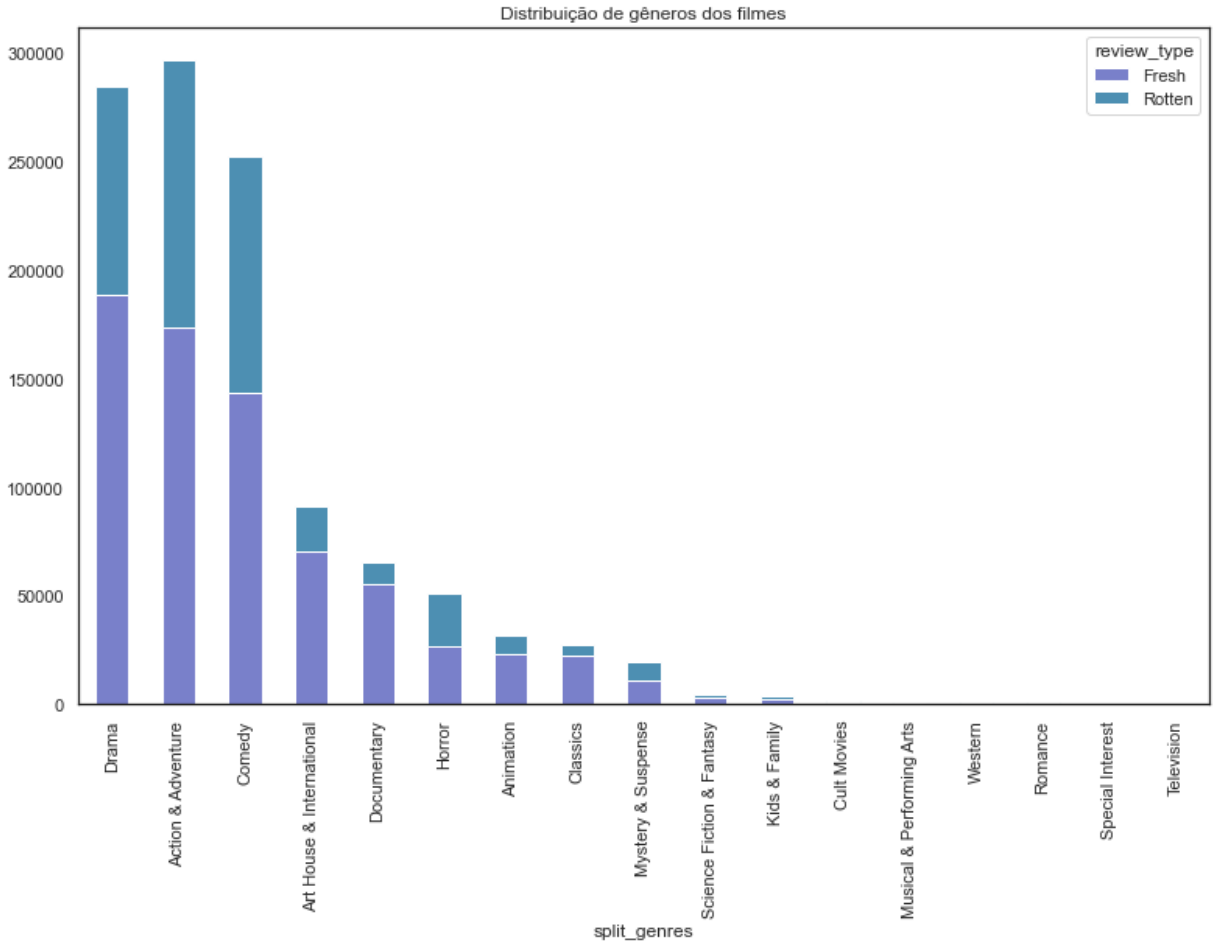


Figura 7: Gênero versus classificação final do filme.

O acervo de filmes é bem extenso, e apesar das críticas serem todas modernas (conforme mostra a figura 9) - visto que o *Rotten Tomatoes* é uma plataforma digital com pouco mais de 20 anos, os filmes contidos lá possuem datas muito mais amplas. Há filmes clássicos, do início do século 20 a filmes do século 21. Na figura 8 observa-se essa distribuição.

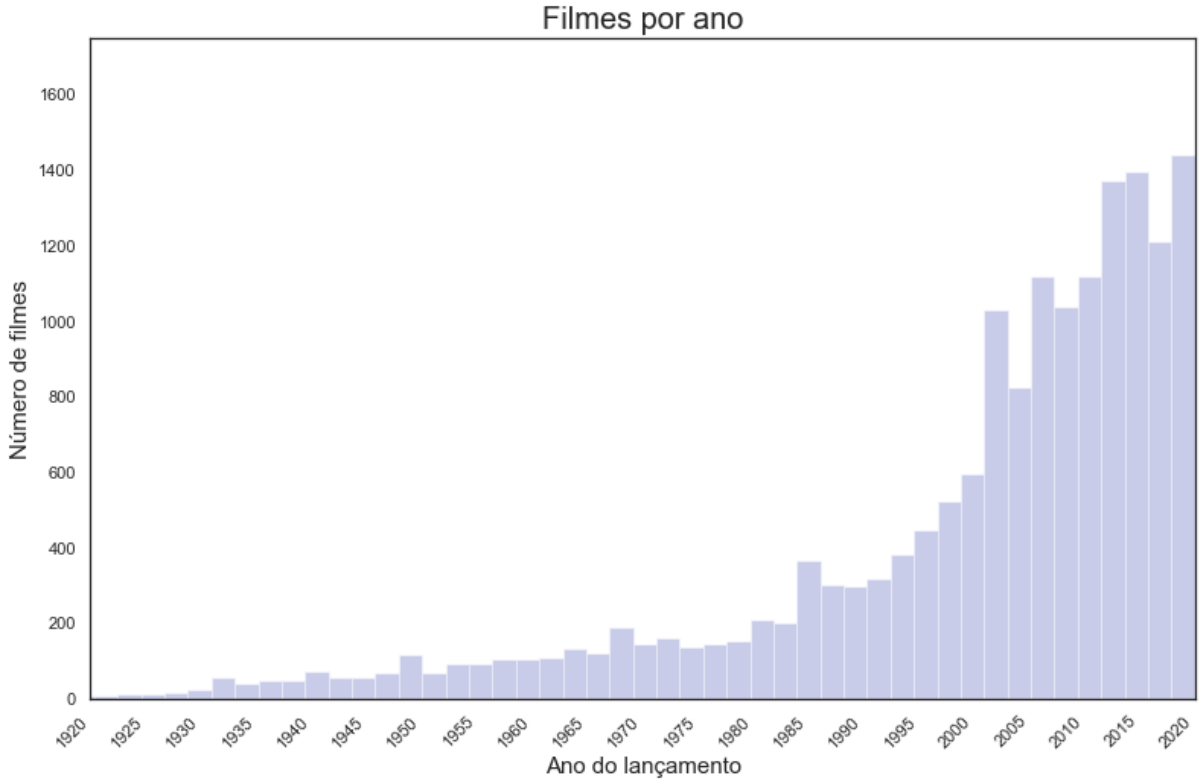


Figura 8: Distribuição dos filmes por ano de lançamento.

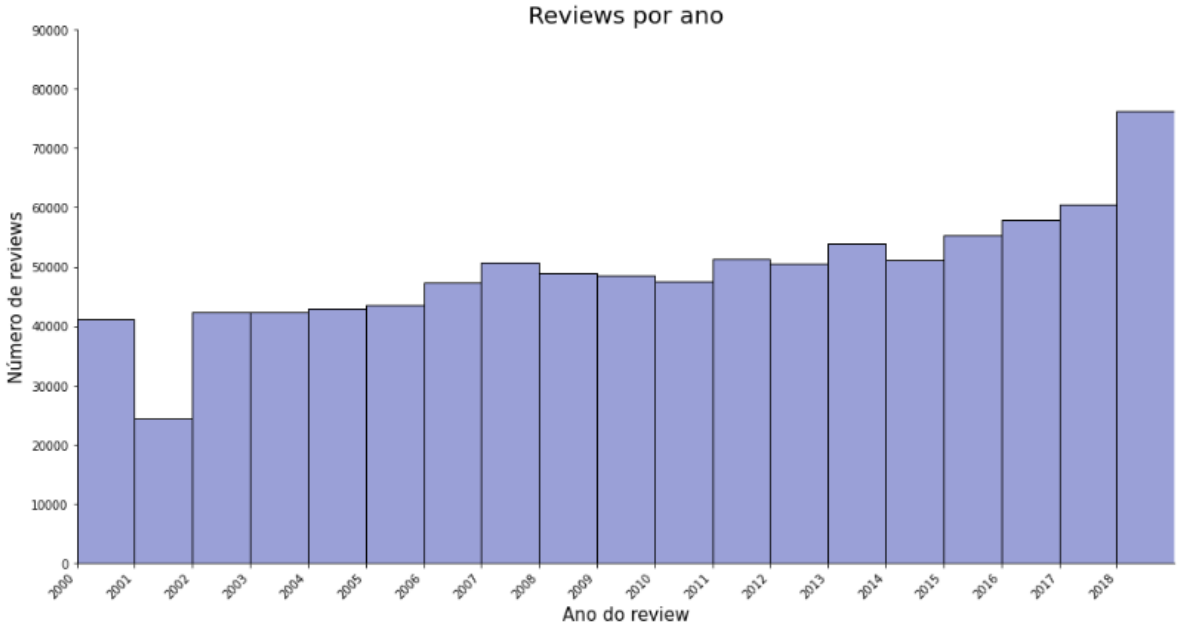


Figura 9: Distribuição dos reviews por ano.

5. Criação de Modelos de Machine Learning

A construção deste trabalho foi feita em uma máquina local utilizando Jupyter Notebook na linguagem Python.

As bibliotecas usadas foram as bibliotecas clássicas de ciência de dados, como Pandas para leitura e manipulação dos conjuntos de dados, Matplotlib e Seaborn para a criação das visualizações e Sklearn para os classificadores.

Para fins de criação do modelo de Machine Learning, foi utilizada a medida de TF-IDF: *Term Frequency - Inverse Document Frequency*, que consiste no cálculo da importância de um termo num documento dentro de um *corpus*, isto é, uma coleção de documentos.

Como dito anteriormente, o objetivo deste trabalho é utilizar os reviews dos filmes com a classificação dos mesmos entre *Fresh* e *Rotten.* Para isso, foi utilizado o conjunto de dados de Review apenas. Neste conjunto há a coluna *review_content*, onde se tem todos os reviews feitos por críticos, e a coluna *review_type,* onde se tem a classificação desse review. Essas duas colunas foram utilizadas para treinar o modelo - a coluna *review_content* como a variável independente, isto é, o X, e a coluna *review_type* como a variável dependente, o y.

Os rótulos estão levemente desbalanceados, mas ainda num patamar aceitável para a classificação:

- Fresh: 681035
- Rotten: 383176

Foi utilizada a técnica de *train_test_split* para dividir o conjunto de dados em dois conjuntos menores, sendo o primeiro com 70% do tamanho original destinado para o treino do modelo e os outros 30% destinados ao teste do mesmo.

```
[44] df = rt_reviews[['review_content', 'review_type']]
```

```
[45] from sklearn.model_selection import train_test_split

     X = df['review_content']
     y = df['review_type']

     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
                                                         random_state=42,
                                                         stratify=df['review_type'])
```

Figura 10: Definição das variáveis dependente e independente e divisão do conjunto de dados entre "teste" e "treino".

Em conjunto com o modelo TF-IDF, foram utilizados outros classificadores a fim de calcular e comparar as performances. Estes são detalhados a seguir.

- Regressão Logística:

```
# Regressão Logística
from sklearn.linear_model import LogisticRegression

class_reglog = Pipeline([('tfidf', TfidfVectorizer()),
                    ('class',  LogisticRegression(solver='lbfgs', max_iter=500))])

class_reglog.fit(X_train, y_train)
y_reglog = class_reglog.predict(X_test)
y_proba_reglog = class_reglog.predict_proba(X_test)
```

Figura 11: Implementação do modelo de Regressão Logística.

Apresentou bons resultados, tanto na acurácia geral quanto nas métricas de classes separadamente:

	precision	recall	f1-score	support
Fresh	0.84	0.89	0.86	204311
Rotten	0.78	0.69	0.74	114953
accuracy			0.82	319264
macro avg	0.81	0.79	0.80	319264
weighted avg	0.82	0.82	0.82	319264

A curva ROC mostra que o modelo tem uma capacidade boa de distinguir a classe Fresh da classe Rotten (figura 12), com uma área sob a curva igual a 0,89.

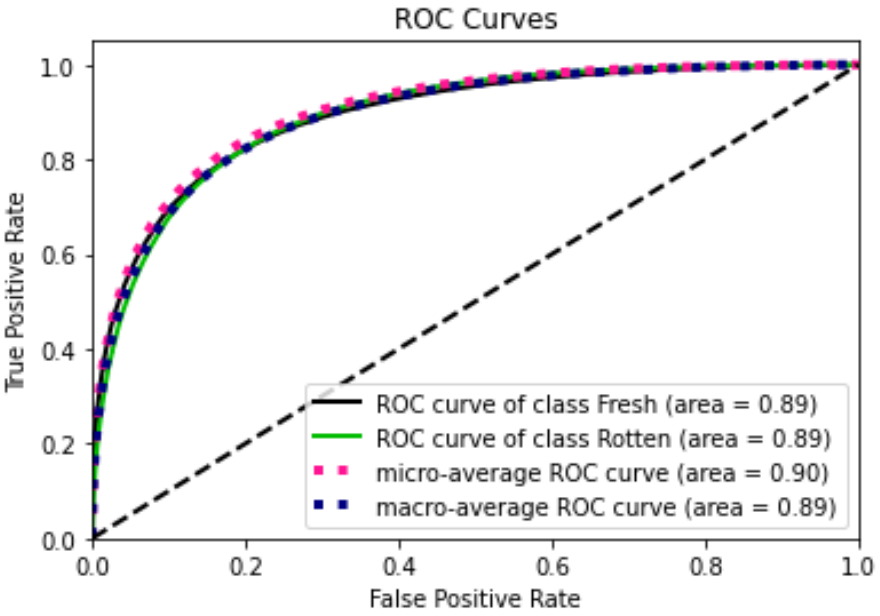


Figura 12: Curva ROC mostrando as áreas sob a curva para as duas classes para o modelo de Regressão Logística.

- Gradient Boosting:

```
# XGBoost
from sklearn.ensemble import GradientBoostingClassifier

class_xgb = Pipeline([('tfidf', TfidfVectorizer()),
                      ('class', GradientBoostingClassifier())])

class_xgb.fit(X_train, y_train)
y_xgb = class_xgb.predict(X_test)
y_proba_xgb = class_xgb.predict_proba(X_test)
```

Figura 13: Implementação do modelo Gradient Boosting.

Apresentou resultados pobres, com uma acurácia baixa e valor de *recall* muito inferior para a classe Rotten, o que mostra que o modelo teve muita dificuldade em classificar registros como Rotten quando eles eram de fato pertencentes a esta classe:

	precision	recall	f1-score	support
Fresh	0.67	0.98	0.80	204311
Rotten	0.81	0.14	0.24	114953
accuracy			0.68	319264
macro avg	0.74	0.56	0.52	319264
weighted avg	0.72	0.68	0.60	319264

- Linear SVC:

```
# SVC
from sklearn.svm import LinearSVC
from sklearn.calibration import CalibratedClassifierCV

svm = LinearSVC()
clf = CalibratedClassifierCV(svm)

class_svc = Pipeline([('tfidf', TfidfVectorizer()),
                      ('class', clf)])

class_svc.fit(X_train, y_train)
y_svc = class_svc.predict(X_test)
y_proba_svc = class_svc.predict_proba(X_test)
```

Figura 14: Implementação do modelo Linear SVC.

Apresentou resultados bons, similares aos resultados da Regressão Linear. A acurácia se manteve a mesma, e as métricas individuais das classes se mostraram muito similares, o que indica que esse modelo tem uma boa capacidade de distinguir uma classe da outra:

	precision	recall	f1-score	support
Fresh	0.84	0.89	0.86	204311
Rotten	0.78	0.70	0.74	114953
accuracy			0.82	319264
macro avg	0.81	0.79	0.80	319264
weighted avg	0.82	0.82	0.82	319264

Com métricas extremamente parecidas com o modelo de Regressão Logística, a curva ROC se apresentou da mesma maneira, com uma área sob a curva de 0.89 (figura 15).

```
skplt.metrics.plot_roc(y_test, y_proba_svc)
plt.figure(figsize=(10,8))
plt.show()
```

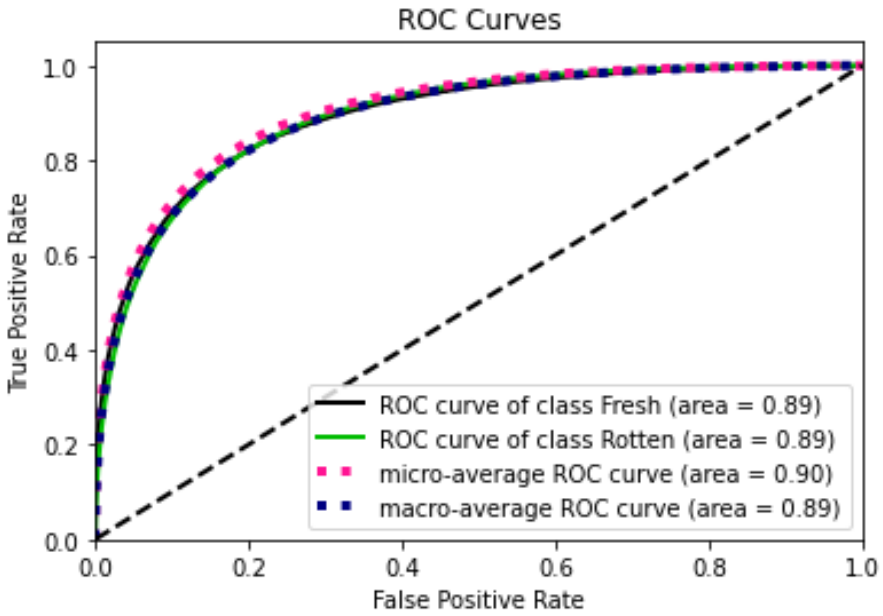


Figura 15: Curva ROC mostrando as áreas sob a curva para as duas classes para o modelo Linear SVC.

- KNN Classifier:

```
from sklearn.neighbors import KNeighborsClassifier

class_knn = Pipeline([('tfidf', TfidfVectorizer()),
                      ('class', KNeighborsClassifier())])

class_knn.fit(X_train, y_train)
y_knn = class_knn.predict(X_test)
y_proba_knn = class_knn.predict_proba(X_test)
```

Figura 16: Implementação do modelo KNN Classifier.

Dos modelos testados, este foi o que apresentou a pior performance, além de demorar tempo superior para rodar. A acurácia deste modelo foi ainda mais baixa que a do modelo Gradient Boosting, apesar de ter demonstrado métricas melhores para as classes individualmente:

	precision	recall	f1-score	support
Fresh	0.68	0.71	0.70	204311
Rotten	0.44	0.39	0.41	114953
accuracy			0.60	319264
macro avg	0.56	0.55	0.56	319264
weighted avg	0.59	0.60	0.59	319264

A curva ROC também apresentou resultados muito pobres, com uma área sob a curva de apenas 0.58, o que indica que esse modelo não tem capacidade de distinguir uma classe da outra e o resultado obtido é praticamente randômico.

```
skplt.metrics.plot_roc(y_test, y_proba_knn)
plt.figure(figsize=(10,8))
plt.show()
```

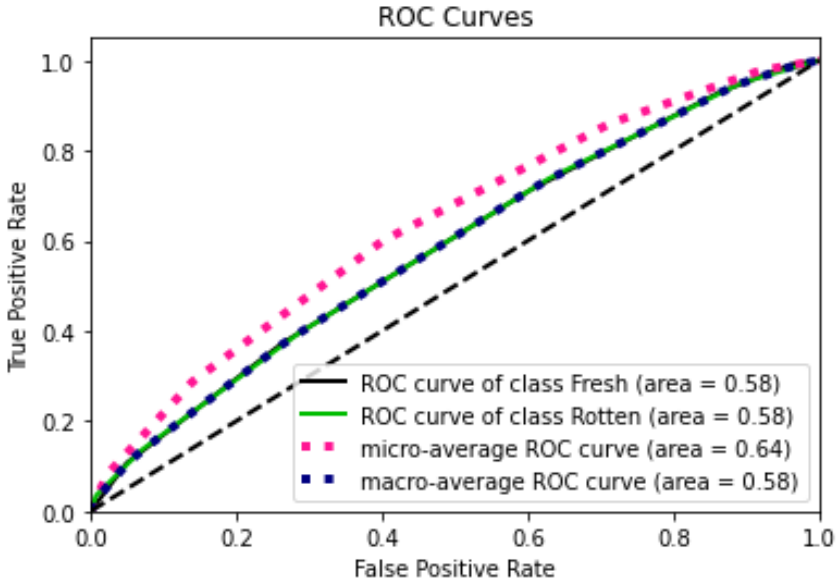


Figura 17: Curva ROC mostrando as áreas sob a curva para as duas classes para o modelo KNN Classifier.

Para fins de comparação dos modelos utilizados, a métrica escolhida foi a da acurácia, por não se tratar de um conjunto de dados com alto nível de desbalanceamento. Essas métricas foram colocadas em uma tabela para melhor visualização e comparação:

Acurácia	Modelo
0,820587	Regressão Logística
0,820622	**Linear SVC**
0,679412	Gradient Boosting
0,599310	KNN Classifier

Como dito anteriormente, os modelos de Regressão Logística e Linear SVC apresentaram resultados extremamente parecidos, porém, através dessa tabela é possível verificar uma ligeira vantagem do segundo em relação ao primeiro.

5.1. Reavaliação dos Parâmetros

Os modelos treinados receberam como parâmetro os textos dos *reviews* no formato em que se encontravam no *dataset*, sem que nenhum outro tipo de pré-processamento fosse feito além da limpeza dos nulos. Isso ocorreu por ser um *dataset* muito extenso - qualquer processamento tomaria tempo e recursos em excesso. Porém, para fins de uma análise mais fina - e para contornar o problema da falta de processamento, foi feita uma amostragem dos dados estratificada por filme e status do review, para que o *dataset* continuasse representativo em relação ao original.

Desse modo, utilizando 40% do *dataset* original, foi possível obter um conjunto de dados com um tamanho mais acessível - em termos de custo de processamento e tempo, com 425.384 entradas.

```
# Reduzindo os dados para poder aplicar a lematização. Dados estratificados por filme e tipo de review.
rt_reviews_strat = rt_reviews.groupby(['rotten_tomatoes_link','review_type'],
                                      group_keys=False).apply(lambda x: x.sample(frac=0.4))
```

Figura 18: Código utilizado para reduzir a base, estratificando por filme e classificação final do filme.

Feita a amostragem, foi necessário verificar se o texto do review estava na língua inglesa. Para tal, foi necessária a criação de uma função, utilizando a biblioteca *langid*, que processou o conjunto de dados para que se pudesse definir a

língua em que se encontrava cada texto. Foram encontrados 420.727 textos em inglês.

```
def lang_idf(df):
    '''Checa o texto para ver se a lingua escrita é inglês, se não for retorna Falso.
    Args:
        df: dataframe object.
    Returns:
        df: dataframe object com coluna indicando a língua.'''
    lang = []
    for text in df['review_content']:
        lang.append(langid.classify(text)[0])
    df['lang_idf'] = lang

    return df
```

Figura 19: Função criada para determinar a língua dos textos de reviews.

Após a limpeza dos textos em língua não inglesa, foi feita a lematização dos dados, utilizando a biblioteca *spacy*.

```
def prepare_text(df):
    '''Lemmatiza o texto para passar pelo modelo.
    Args:
        df: dataframe object.
    Returns:
        df: dataframe object com o texto preparado.'''

    mov_end = []

    for index, item in enumerate(nlp.pipe(df['review_content'], n_process=6, batch_size=2000)):
        doc = nlp(item)
        lemmatized_sentence = " ".join([token.lemma_ for token in doc if not token.is_stop])
        mov_end.append(lemmatized_sentence)

    df['prepared_review_content'] = mov_end

    return df
```

Figura 20: Função criada para lematizar os textos de reviews utilizando o pipeline da biblioteca spacy.

A lematização, de acordo com a Universidade de Stanford[3], consiste em reduzir as formas flexionadas e derivadas das palavras a uma base comum, por exemplo:

am, are is = be

car, cars, car's = car

[3] https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html

Os dados, então, foram processados utilizando os mesmos algoritmos anteriormente usados. Surpreendentemente, todos os modelos se comportaram ligeiramente pior nos dados limpos e lematizados. Os resultados, porém, foram similares em relação à performance dos algoritmos quando comparados entre si: Regressão Logística e Linear SVC performaram melhor enquanto que Gradient Boosting e KNN tiveram performance não satisfatória, sendo o último o pior de todos. Nesse cenário, ao contrário, observa-se que a Regressão Logística obteve a melhor performance - no cenário anterior Linear SVC havia apresentado o melhor resultado. Apesar disso, optou-se por utilizar este modelo de Regressão Logística para aplicar ao conjunto de dados dos *reviews* dos usuários.

Acurácia	**Modelo**
0,796798	**Regressão Logística**
0,793494	Linear SVC
0,677727	Gradient Boosting
0,495876	KNN Classifier

5.2. Avaliação do Modelo no Conjunto de Dados de *Reviews* de Usuários

Usando os dados do filme “Mãe!”, que foram raspados do site do *Rotten Tomatoes*, foi necessário adequá-los à forma como o algoritmo foi treinado: em inglês e lematizados.

Das 3.768 entradas de *reviews* obtidas, após a verificação do idioma, restaram 3.544.

Aplicado o modelo de Regressão Linear, a classificação dada pelo modelo aos reviews foi salva na coluna “clas_pred”. Observa-se o resultado:

- Rotten: 1941
- Fresh: 1603

Dos 3.544 reviews, 54,8% dos textos de *review* dos usuários foram classificados como *Rotten.*

É possível fazer uma checagem ao ver as primeiras linhas do resultado:

	rotten_tomatoes_link	review_content	lang_idf	prepared_review_content	class_pred
0	m/mother_2017	An intense, gorgeous and brilliant allegory. I...	en	intense , gorgeous brilliant allegory . little...	Fresh
1	m/mother_2017	too literal. lacks any kind of nuance. lacks c...	en	literal . lack kind nuance . lack creativity e...	Rotten
2	m/mother_2017	As though as it is to watch and understand, it...	en	watch understand , brilliantly direct think .	Fresh
3	m/mother_2017	This movie had me pissed off for 2 hours. Thou...	en	movie piss 2 hour . thought ending massively g...	Rotten
4	m/mother_2017	Mother! Is frustrating, stupidly surrealistic ...	en	mother ! frustrating , stupidly surrealistic d...	Fresh
6	m/mother_2017	The most remarkable thing about this film is J...	en	remarkable thing film Jennifer Lawrence perfor...	Fresh
7	m/mother_2017	This is very strange, confused me, in general ...	en	strange , confuse , general know like .	Rotten
8	m/mother_2017	Everything in this movie is symbolic. You thin...	en	movie symbolic . think go sense , . movie stre...	Rotten
9	m/mother_2017	It is WILD ride, while also depressing.	en	wild ride , depressing .	Fresh
10	m/mother_2017	Fuck it. They are playing with our emotions. W...	en	fuck . play emotion . bad movie see entire lif...	Rotten

Figura 21: Dataset de reviews dos usuários com a coluna "class_pred", predita pelo modelo.

A partir dessa amostra, ao se fazer a leitura dos *reviews* e compará-los com a classificação dada pelo modelo, pode-se concluir que o algoritmo pôde classificar corretamente esses *reviews*.

Pode-se também comparar o percentual de críticos e usuários que avaliaram positivamente este filme:

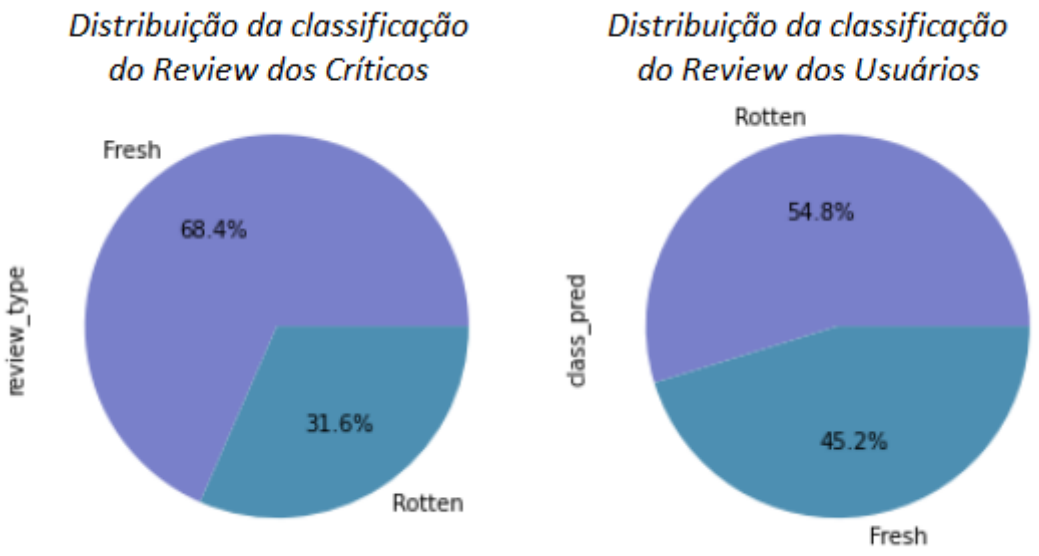


Figura 22: Gráficos comparando a distribuição da classificação entre Fresh e Rotten pelos críticos e pelos usuários.

Observa-se aqui que grande parte dos críticos avaliou de forma positiva, ou seja, o classificou como *Fresh.* Em contrapartida, mais da metade dos usuários deixaram *reviews* negativos, o que levou o modelo a classificá-los como *Rotten.*

6. Apresentação dos Resultados

De acordo com a sugestão de desenvolvimento deste trabalho, foi criado um Canvas *workflow* seguindo o modelo proposto. Esse *workflow* pode ser visto em maior detalhe no apêndice.

Fresco ou Podre: uma análise dos *reviews* do *Rotten Tomatoes*

AUTORA

Bárbara Gonçalves Oliveira, aluna de pós-graduação da PUC Minas.

1

DEFINIÇÃO DO PROBLEMA

Examinar *reviews* de filmes a fim de fazer uma classificação e tirar uma conclusão final sobre se o filme é bom ou ruim em termos gerais.

2

RESULTADOS E PREDIÇÕES

Classificador: Possibilitar a classificação correta dos *reviews* de críticos entre "*Rotten*" e "*Fresh*".
Espera-se um modelo capaz de aferir corretamente a classificação fornecida pelo crítico dado o teor do texto.

Objetivo: Obter a capacidade de classificar um filme entre *Rotten* e *Fresh* dado um texto de revisão qualquer.

3

AQUISIÇÃO DOS DADOS

Dados obtidos do *Rotten Tomatoes* a partir de duas fontes distintas:

Dados dos filmes e dos *reviews* dos críticos coletados da plataforma *Kaggle*.

Dados dos *reviews* de usuários coletados através de uma raspagem de dados do próprio site do *Rotten Tomatoes*.

4

MODELAGEM

Algoritmos de aprendizado de máquina supervisionado.

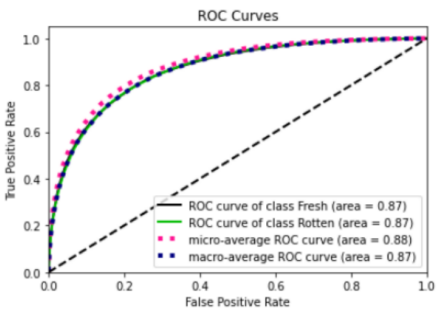
Foram feitas análises utilizando os algoritmos clássicos da biblioteca sklearn:

Regressão Logística, Gradient Boosting, Linear SVC e KNN.

5

AVALIAÇÃO DO MODELO

Relatórios de Classificação, olhando para a acurácia geral, os índices de previsão e revocação das classes eparadamente.
A área sob a curva ROC também foi uma das métricas importantes para a definição do modelo.



6

PREPARAÇÃO DOS DADOS

Remoção de dados nulos, verificação do idioma dos textos e limpeza dos não-ingleses, redução das palavras dos textos a seus termos comuns utilizando a técnica de lematização.

ATIVAÇÃO (VIDE REPOSITÓRIO DO GITHUB)

https://github.com/BarbaraOlive/rotten_tomatoes_classifier

O desenvolvimento deste trabalho possibilitou a avaliação e classificação do filme usado como exemplo, mas não se limita somente a ele. É possível conduzir esta análise para qualquer outro filme que se queira - desde que este esteja disponível no site do *Rotten Tomatoes* e tenha *reviews* de usuários.

Ainda, este projeto foi elaborado utilizando de algoritmos de aprendizado de máquina que foram treinados para fazer um trabalho que seria no mínimo tedioso e muito demorado para um ser humano: ler todos os *reviews* deixados pelos usuários da plataforma *Rotten Tomatoes*, avaliar se são positivos ou negativos e comparar com a classificação dada pelos críticos.

Desta forma, é como se se pudesse ensinar a máquina a ler e a compreender o teor do texto. Isso foi feito mostrando para ela exemplos suficientes de textos e seus respectivos rótulos de classificação. Assim, possibilita-se que ela identifique padrões e tome decisões baseadas nesses padrões.

Para ilustrar ainda mais essa análise, pode-se fazer a apresentação dos textos dos *reviews* utilizando a nuvem de palavras - que consiste em uma representação visual dos termos de um texto e é baseada na frequência das palavras - quanto mais frequente, maior o tamanho dela.

Para o filme em questão, a nuvem de palavras dos *reviews* dos críticos, sem distinção da classificação (vide figura 23), mostra diversos termos que podem ser considerados positivos ou negativos. Há termos como *experience, allegory, horror, bad, think, great* e *love.*

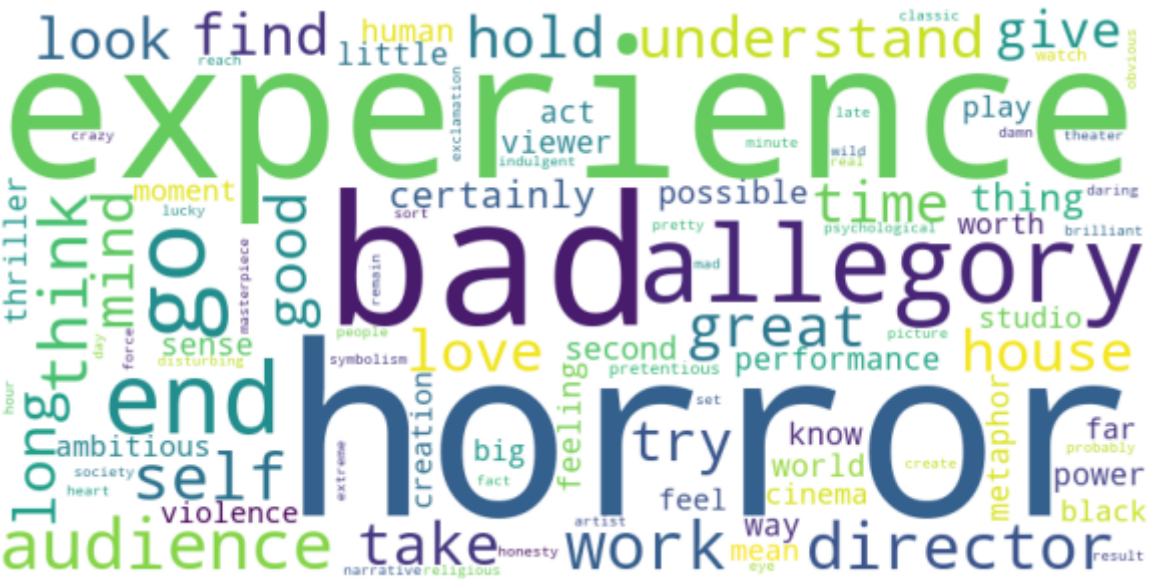


Figura 23: Nuvem de palavras com os reviews dos críticos para o filme "Mãe!"

Porém, é possível estreitar essa análise gerando uma nuvem de palavras apenas para os *reviews* classificados pelos críticos como *Rotten* (figura 24) e outra para os classificados como *Fresh* (figura 25). Após a separação, pode-se verificar para a nuvem de palavras dos *reviews* do tipo *Rotten* um enorme termo *bad,* isso se dá pois a lematização reduz os termos derivativos à forma mais simples, como é o caso na figura 26 - o termo *worst* (horrível) é reduzido a *bad*.

Para a nuvem de palavras das classificações do tipo *Fresh* se observa palavras como *love, experience* e *brilliant,* que são palavras positivas dentro desse contexto.

Figura 24: Nuvem de palavras com os reviews dos críticos do tipo Rotten para o filme "Mãe!"

Figura 25: Nuvem de palavras com os reviews dos críticos do tipo Fresh para o filme "Mãe!"

	review_content	prepared_review_content
69	I hated it. It was one of the worst horror mov...	hate . bad horror movie watch , pointless , bo...
2061	Worst Movie Ever ! no plot, pretencious, try t...	bad Movie ! plot , pretencious , try deep end ...
2067	Worst movie of all time. Almost tied with Noah...	bad movie time . tie Noah ... ending happen . ...
2070	Worst movie i ve ever seen! The end of Jennif...	bad movie ve see ! end Jennifer Lawrence car...

Figura 26: Tabela mostrando a redução dos termos derivativos a um termo comum.

Também para o conjunto de dados dos *reviews* dos usuários pode-se gerar a nuvem de palavras e obter os termos mais frequentes, porém, nesse caso, as nuvens geradas foram apenas para os rótulos separados - uma para os *reviews* classificados pelo modelo como *Rotten* (figura 27) e outra para os classificados como *Fresh* (figura 28).

Figura 27: Nuvem de palavras com os reviews dos usuários classificados pelo modelo como do tipo Rotten para o filme "Mãe!"

Figura 28: Nuvem de palavras com os reviews dos usuários classificados pelo modelo como do tipo Fresh para o filme "Mãe!"

Para os que o modelo classificou como *Rotten* temos a presença de termos como *waste time*, *bad, bad movie, terrible* e *awful*, que indicam que realmente esses reviews representam que estes usuários avaliaram negativamente o filme - o que é condizente com a classificação *Rotten*. Já para os classificados pelo modelo como

Fresh aparecem palavras como *good, think, like* e *love,* que indicam também uma correta classificação do modelo.

A partir dessas análises, pode-se concluir que o algoritmo desenvolvido atingiu seu objetivo: ser um termômetro nas análises e avaliações de filmes, entregando informações relevantes e uma boa classificação textual de *reviews*. Com isso, é possível inferir e sumarizar a qualidade de um filme nas perspectivas dos usuários em geral e traçar um comparativo com as opiniões dos críticos.

7. Links

Todos os dados, *Jupyter Notebooks*, arquivos gerados e modelos treinados podem ser encontrados no repositório do Github assim como o infográfico e o Canvas *workflow.*

- Link para o repositório do Github: https://github.com/BarbaraOlive/rotten_tomatoes_classifier
- Link para o vídeo da apresentação do projeto:
- https://youtu.be/BBC02Lw7XOc

Fresco ou Podre: uma análise dos *reviews* do *Rotten Tomatoes*

AUTORA

Bárbara Gonçalves Oliveira, aluna de pós-graduação da PUC Minas.

1

DEFINIÇÃO DO PROBLEMA

Examinar *reviews* de filmes a fim de fazer uma classificação e tirar uma conclusão final sobre se o filme é bom ou ruim em termos gerais.

2

RESULTADOS E PREDIÇÕES

Classificador: Possibilitar a classificação correta dos *reviews* de críticos entre "*Rotten*" e "*Fresh*".
Espera-se um modelo capaz de aferir corretamente a classificação fornecida pelo crítico dado o teor do texto.

Objetivo: Obter a capacidade de classificar um filme entre *Rotten* e *Fresh* dado um texto de revisão qualquer.

3

AQUISIÇÃO DOS DADOS

Dados obtidos do *Rotten Tomatoes* a partir de duas fontes distintas:

Dados dos filmes e dos *reviews* dos críticos coletados da plataforma *Kaggle.*

Dados dos *reviews* de usuários coletados através de uma raspagem de dados do próprio site do *Rotten Tomatoes.*

4

MODELAGEM

Algoritmos de aprendizado de máquina supervisionado.

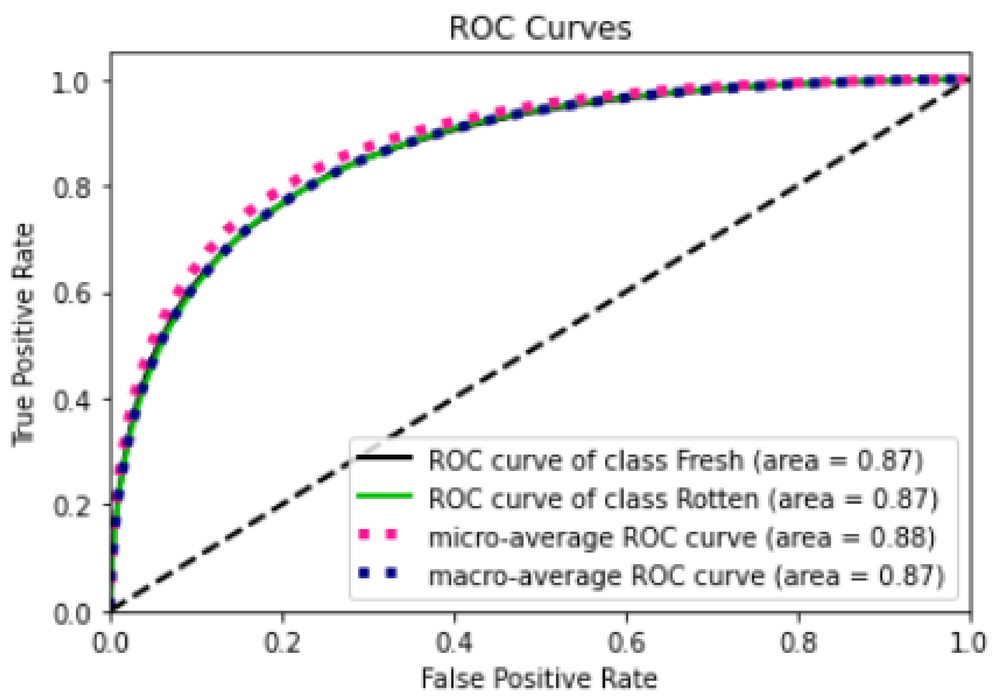
Foram feitas análises utilizando os algoritmos clássicos da biblioteca sklearn:

Regressão Logística, Gradient Boosting, Linear SVC e KNN.

5

AVALIAÇÃO DO MODELO

Relatórios de Classificação, olhando para a acurácia geral, os índices de previsão e revocação das classes eparadamente.
A área sob a curva ROC também foi uma das métricas importantes para a definição do modelo.



6

PREPARAÇÃO DOS DADOS

Remoção de dados nulos, verificação do idioma dos textos e limpeza dos não-ingleses, redução das palavras dos textos a seus termos comuns utilizando a técnica de lematização.

ATIVAÇÃO (VIDE REPOSITÓRIO DO GITHUB)

https://github.com/BarbaraOlive/rotten_tomatoes_classifier

Fresco ou Podre?

uma análise dos *reviews* do *Rotten Tomatoes*

Analisar *reviews* de críticos de filmes e aplicar sobre reviews de usuários para obter uma classificação destes.

DE ONDE VIERAM OS DADOS?

Kaggle: Dados dos filmes e dos *reviews* dos usuários.
Rotten Tomatoes: dados do filme a ser analisado.

FORMATO DOS DADOS

Dados de *reviews* dos críticos: texto em linguagem natural, majoritariamente em língua inglesa.

Rótulo dos *reviews*: "*Rotten*" ou "*Fresh*".

Dados dos *reviews* dos usuários: texto em linguagem natural em diversas línguas. Dados sem rótulos.

Distribuição dos rótulos nos dados de reviews dos críticos.

METODOLOGIAS

Aprendizado de máquina supervisionado
TF-IDF
Classificadores

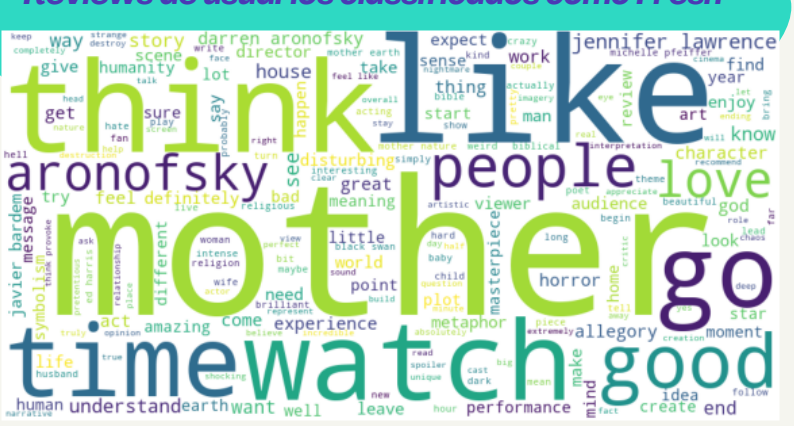
Utilizar os textos em linguagem natural para treinar os modelos.

Lematizar e manter apenas os textos em inglês para treinar os modelos.

RESULTADOS

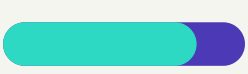
Um modelo treinado capaz de, dado um texto de *review* qualquer, classificá-lo como *Rotten* ou *Fresh*.

Reviews de usuários classificados como Fresh

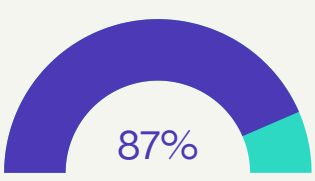


Reviews de usuários classificados como Rotten

*Acurácia**



*Área sob a curva**



** da Regressão Logística, o modelo escolhido*

Referências:
https://github.com/BarbaraOlive/rotten_tomatoes_classifier