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

Victor Hugo Benedetti Delaiba

Análise de sentimentos em comentários no IMDb

Belo Horizonte
2020

Victor Hugo Benedetti Delaiba

Análise de sentimentos em comentários no IMDb

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
2020

SUMÁRIO

1 Introdução

1.1 Contextualização

A quantidade de dados está crescendo exponencialmente ao longo dos últimos anos, e um dos grandes problemas é a dificuldade de extrair informação/conhecimento desses dados. Uma parte relevante desse crescimento é a quantidade de texto gerada em sites, redes sociais, artigos, blogs, fóruns, entre outros. Logo, o processamento de linguagem natural (NLP) se torna cada vez mais importante para compreender automaticamente os textos e transformá-los em informação/conhecimento. Existem várias aplicações que utilizam dessas técnicas, como por exemplo, sumarização de texto, segmentação de tópicos, chatbots e tradução.

Uma das áreas de NLP é a classificação de documentos, que visa prever textos em categorias de acordo com o seu conteúdo e dentro dessa área há a análise de sentimentos, que visa identificar a opinião em um determinado texto, se ela é positiva ou negativa.

A análise de sentimentos pode ser utilizada de diversas formas, principalmente na aplicação em inteligência de negócios, utilizando sites especializados em reviews de clientes ou redes sociais, buscando um feedback do público geral a respeito de algum produto ou serviço.

1.2 O problema proposto

O problema que será abordado é a análise de sentimentos dos comentários de usuários do site IMDB sobre os títulos de produções cinematográficas. O objetivo é saber a percepção dos usuários em relação a esses títulos por meio do que está escrito nos comentários. Pode-se utilizar esta análise para diversas finalidades, como por exemplo, o próprio IMDb poderia utilizar para recomendar títulos para os usuários, produtoras poderiam avaliar a receptividade ao lançarem um filme, um dono de cinema poderia planejar a demanda pelo filme ao ver a reação dos usuários.

Cabe ressaltar que todos os comentários estão escritos na língua inglesa.

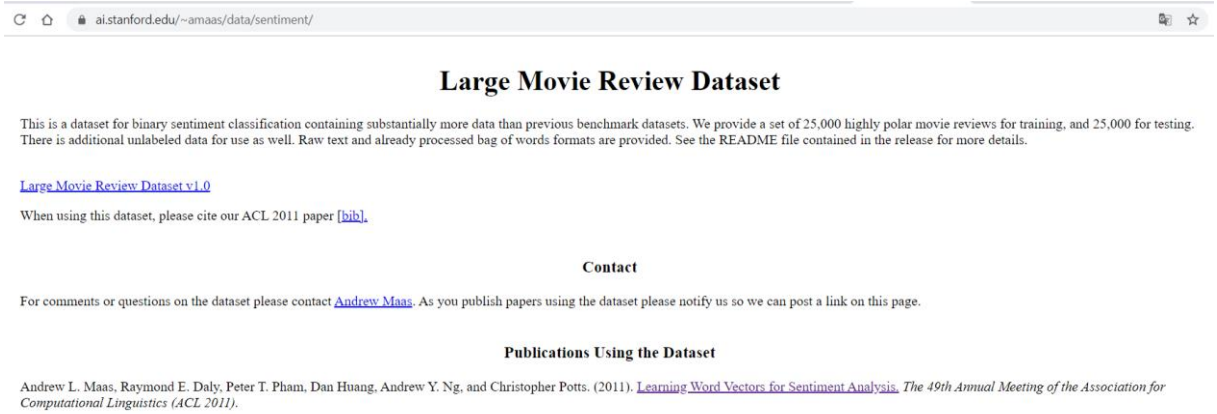
1.3 Estrutura do trabalho

O trabalho está estruturado em seis partes. A introdução apresenta a contextualização do assunto tratado e o problema proposto. Na sequência, são apresentadas as informações acerca da coleta dos dados (Capítulo 2) e os passos realizados para o tratamento (Capítulo 3). No capítulo 4 são efetuadas a análise e a exploração desses dados, com vistas à obtenção de informações estatísticas relevantes. Como o trabalho gira em torno de análise textual, o Capítulo 5 – Aplicação de Modelos de Machine Learning – abrange o processamento de linguagem natural utilizando algumas técnicas. Por fim, no Capítulo 6 são apresentados os resultados obtidos e a conclusão.

2 Coleta de Dados

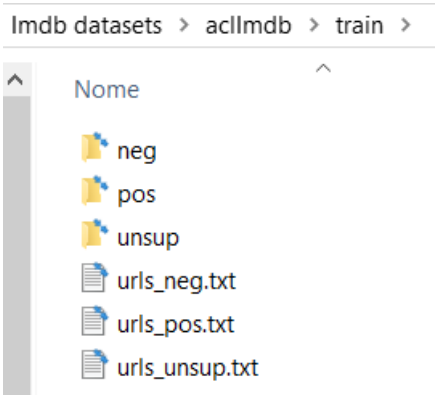
Os dados dos comentários do IMDB foram obtidos no endereço https://ai.stanford.edu/~amaas/data/sentiment/, no dia 23/07/2020. Esse dataset contém 100.000 comentários de usuários relacionados a url do título no site do imdb. Esse conjunto está dividido em 25.000 dados anotados (com a classificação de sentimento feita por humano) para treinamento e 25.000 dados anotados para teste e 50.000 dados não anotados.

Figura 1 - Site da Stanford que disponibilizou o dataset



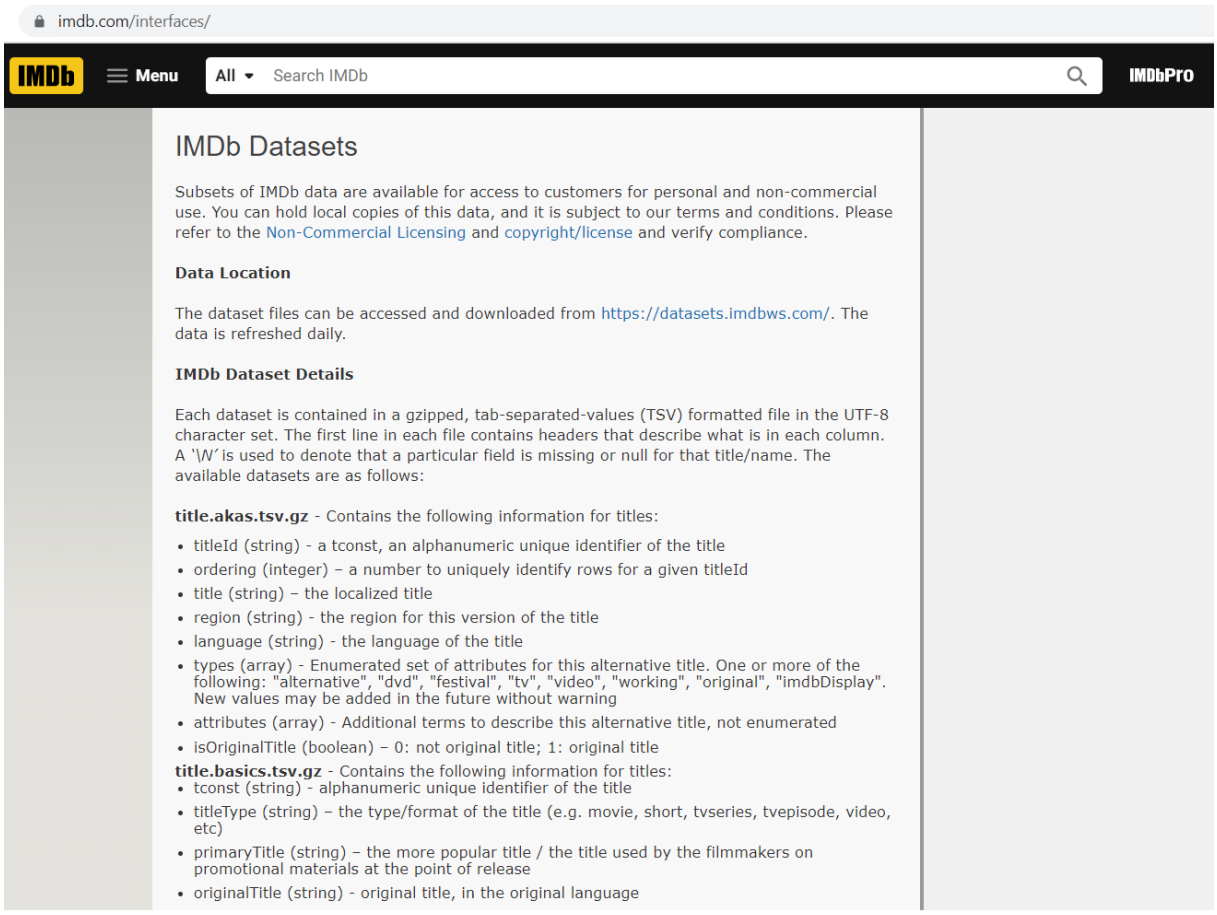
Os dados são disponibilizados em formato texto. O dataset é separado com os dados de treinamento e teste. Dentro da pasta de treinamento (figura 2), os comentários são separados por positivos, negativos e não anotados. Além disso, existem os arquivos de urls, separados da mesma maneira, que informam qual o endereço no site do IMDb do título de determinado comentário. Dentro desses diretórios, cada comentário é representado por um arquivo em formato txt. Os dados de teste têm a mesma estruturação, exceto por não conter os dados não anotados.

Figura 2 - Estrutura dos arquivos de treinamento



Além dos comentários, foram coletadas informações, no formato TSV (tab-separated-values), sobre esses títulos no site https://www.imdb.com/interfaces/, conforme demonstra a figura 3:

Figura 3 - Site do IMDb que disponibiliza informações dos títulos



Esta interface disponibiliza várias informações sobre os títulos em tabelas, porém optou-se por utilizar apenas duas delas, conforme o detalhamento a seguir:

Tabela 1 - Dados básicos (arquivo title.basics.tsv.gz)

Nome da coluna/campo	*Descrição*	*Tipo*
tconst	Identificador do título de filme, usando para fazer o relacionamento entre o outro arquivo e os comentários	Cadeia de caracteres
titleType	O tipo/formato do título (ex. filme, curta-metragem, séries de tv, etc)	Cadeia de caracteres
primaryTitle	O título mais comum	Cadeia de caracteres

originalTitle	Título original, na língua de origem	Cadeia de caracteres
startYear	Representa o ano de lançamento do título	Numérico (quarto dígitos representando o ano)
endYear	Ano da última temporada para series de tv. '\N' para todos os outros tipos.	Numérico (quarto dígitos representando o ano)
runtimeMinutes	Tempo de duração do título	Numérico
genres	Gênero do título (ex: terror, drama, comédia, etc)	Cadéia de caracteres

Tabela 2 - Dados de avaliação (arquivo: title.rantings.tsv.gz)

Nome da coluna/campo	Descrição	Tipo
tconst	Identificador do título de filme, usando para fazer o relacionamento entre o outro arquivo e os comentários	Cadeia de caracteres
averageRating	Média da avaliação de todos os usuários.	Numérico
numVotes	Número de votos para o título	Numérico

3 Processamento/Tratamento de Dados

O primeiro passo do processamento foi identificar os títulos citados nos comentários por meio dos arquivos de urls, pois é possível extrair o identificador do mesmo e associá-lo aos das tabelas de dados básicos e de avaliação.

Neste passo, conforme script da figura 4, foi verificado que existiam alguns títulos das urls que não possuíam nenhum título associado nas tabelas de dados básico e avaliação. Porém, quando foi feita a consulta da url em um navegador, o site IMDB redirecionou para uma outra url, ou seja, o identificador *tconst* foi alterado entre a extração da base de comentários e a base atual de dados dos títulos.

Figura 4 - Leitura dos dados e verificação da inexistência de alguns títulos entre as bases

```
urls = open(url_urls, 'r').read()
urls_df = urls.split('\n')
urls_df = pd.DataFrame(urls_df, columns=['urls'])
urls_df = urls_df.loc[urls_df.urls != '']
urls_df['id'] = urls_df['urls'].apply(get_id)

title_basics = pd.read_csv(url_basics, sep='\t', header=0)
title_ratings = pd.read_csv(url_ratings, sep='\t', header=0)

df_basics = urls_df.merge(title_basics, left_on='id', right_on='tconst')

#verificar identificadores que não foram relacionados entre as bases
erros = set(urls_df.id) - set(df_basics.id)
```

Desta maneira, dentro do script de extração dos dados básicos e avaliação, foi necessário criar um passo adicional caso não fosse encontrado o identificador, fazendo o script acessar o site do IMDB (por meio do método GET) e pegar o novo identificador (figura 5).

Figura 5 - Script para extrair o novo código identificador de um título

```
DE, PARA = [], []
for id_ in erros:
    url = 'http://www.imdb.com/title/' + id_
    response = get(url)
    index_new_id = response.text.find('app-argument')
    new_id = response.text[index_new_id + 27:index_new_id + 36]
    DE.append(id_)
    PARA.append(new_id)
```

Em seguida, foi possível realizar a junção entre as bases contemplando todos os títulos da base de comentários. Abaixo, segue a tabela gerada, contendo o identificador dos títulos (*tconst*), o novo identificador caso ele tenha sido modificado e todos os dados básicos:

Figura 6 - Tabela com a junção dos dados básicos e ranking

urls	id	id_new	tconst	titleType	primaryTitle	originalTitle	isAdult	startYear	endYear	runtimeMinutes	genres
http://www.imdb.com/title/tt0406816/usercomments	tt0406816	tt0406816	tt0406816	movie	The Guardian	The Guardian	0	2006	\N	139	Action,Adventure,Drama
http://www.imdb.com/title/tt0406816/usercomments	tt0406816	tt0406816	tt0406816	movie	The Guardian	The Guardian	0	2006	\N	139	Action,Adventure,Drama
http://www.imdb.com/title/tt0406816/usercomments	tt0406816	tt0406816	tt0406816	movie	The Guardian	The Guardian	0	2006	\N	139	Action,Adventure,Drama
http://www.imdb.com/title/tt0406816/usercomments	tt0406816	tt0406816	tt0406816	movie	The Guardian	The Guardian	0	2006	\N	139	Action,Adventure,Drama
http://www.imdb.com/title/tt0406816/usercomments	tt0406816	tt0406816	tt0406816	movie	The Guardian	The Guardian	0	2006	\N	139	Action,Adventure,Drama
http://www.imdb.com/title/tt0406816/usercomments	tt0406816	tt0406816	tt0406816	movie	The Guardian	The Guardian	0	2006	\N	139	Action,Adventure,Drama
http://www.imdb.com/title/tt0085461/usercomments	tt0085461	tt0085461	tt0085461	movie	The Dresser	The Dresser	0	1983	\N	118	Drama
http://www.imdb.com/title/tt0065611/usercomments	tt0065611	tt0065611	tt0065611	movie	Darling Lili	Darling Lili	0	1970	\N	136	Comedy,Drama,Musical
http://www.imdb.com/title/tt0065611/usercomments	tt0065611	tt0065611	tt0065611	movie	Darling Lili	Darling Lili	0	1970	\N	136	Comedy,Drama,Musical
http://www.imdb.com/title/tt0065611/usercomments	tt0065611	tt0065611	tt0065611	movie	Darling Lili	Darling Lili	0	1970	\N	136	Comedy,Drama,Musical
http://www.imdb.com/title/tt0065611/usercomments	tt0065611	tt0065611	tt0065611	movie	Darling Lili	Darling Lili	0	1970	\N	136	Comedy,Drama,Musical
http://www.imdb.com/title/tt0065611/usercomments	tt0065611	tt0065611	tt0065611	movie	Darling Lili	Darling Lili	0	1970	\N	136	Comedy,Drama,Musical
http://www.imdb.com/title/tt0065611/usercomments	tt0065611	tt0065611	tt0065611	movie	Darling Lili	Darling Lili	0	1970	\N	136	Comedy,Drama,Musical
http://www.imdb.com/title/tt0065611/usercomments	tt0065611	tt0065611	tt0065611	movie	Darling Lili	Darling Lili	0	1970	\N	136	Comedy,Drama,Musical
http://www.imdb.com/title/tt0103886/usercomments	tt0103886	tt0103886	tt0103886	tvMovie	The Broken Cord	The Broken Cord	0	1992	\N	92	Drama
http://www.imdb.com/title/tt0184474/usercomments	tt0184474	tt0116308	tt0116308	movie	Fire	Fire	0	1996	\N	108	Drama,Romance
http://www.imdb.com/title/tt0762073/usercomments	tt0762073	tt0762073	tt0762073	movie	Thirst	Bakjwi	0	2009	\N	134	Drama,Fantasy,Horror

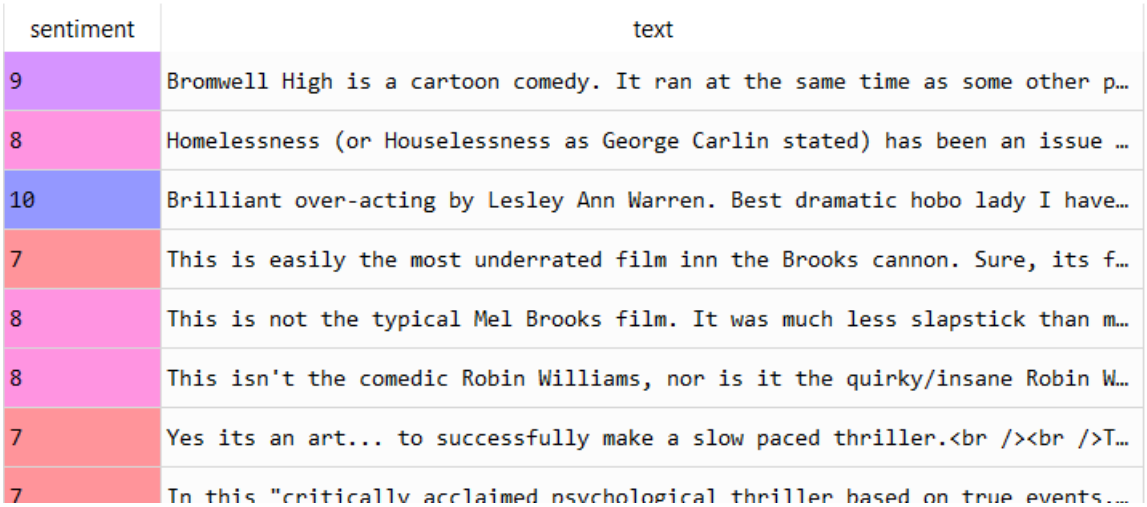
O próximo passo foi extrair o conteúdo textual dos comentários. Para isso, foi feita a leitura de cada arquivo dentro das pastas para transformá-los numa tabela.

Figura 7 - Script para extrair os dados textuais em um dataframe pandas

```
for filename in os.listdir(directory):
    # print(filename)
    if filename.endswith(".txt"):
        f = open(directory + '/' + filename, encoding="utf8")
        lines = f.read()
        comments = comments.append({'index': filename.split('_')[0],
                                    'sentiment': filename.split('_')[1].split('.')[0],
                                    'text': lines}, ignore_index=True)
```

Na figura abaixo, é apresentado o resultado da extração, que contém o comentário e o respectivo valor do sentimento do usuário. Cabe destacar que são considerados negativos os que recebem nota de 0 a 3, e, positivos, de 7 a 10 .

Figura 8 - Resultado da extração dos arquivos textuais positivos



sentiment	text
9	Bromwell High is a cartoon comedy. It ran at the same time as some other p…
8	Homelessness (or Houselessness as George Carlin stated) has been an issue …
10	Brilliant over-acting by Lesley Ann Warren. Best dramatic hobo lady I have…
7	This is easily the most underrated film inn the Brooks cannon. Sure, its f…
8	This is not the typical Mel Brooks film. It was much less slapstick than m…
8	This isn't the comedic Robin Williams, nor is it the quirky/insane Robin W…
7	Yes its an art... to successfully make a slow paced thriller. T…
7	In this "critically acclaimed psychological thriller based on true events,…

Por fim, foi necessário juntar os dois *datasets* gerados nos passos anteriores, o de dados positivos e o de negativos, para termos todos os dados em um único *dataset*.

Ressalta-se que não foram encontrados registros duplicados. Foram identificados dados nulos na coluna *EndYear*, mas isso já era esperado, uma vez que essa informação só se aplica para séries de TV. Também foram encontrados gêneros (*genres*) nulos no conjunto de teste e não foi preciso fazer nenhum preenchimento desse dado.

4 Análise e Exploração dos Dados

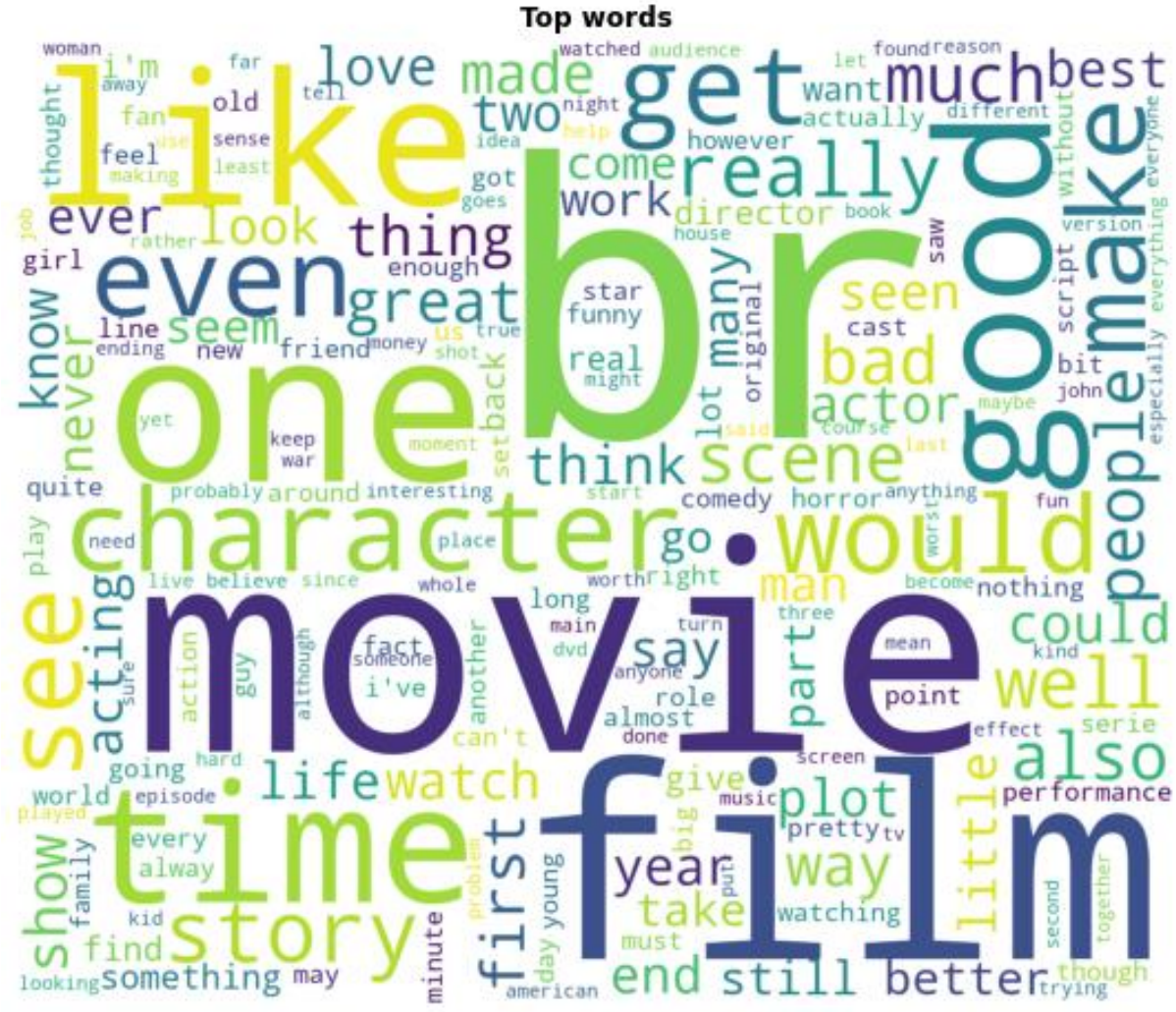
4.1 Análise do conteúdo textual dos comentários

A primeira etapa foi analisar as palavras mais comuns utilizadas nos comentários, para isso criou-se uma nuvem de palavras (figura 10) usando o pacote WordCloud, conforme script da figura 9.

Figura 9 - Script para geração da nuvem de palavras

```
#%% word cloud
stopwords = nltk.corpus.stopwords.words('english')

text = ' '.join(df_train['text'].str.lower().values)
wordcloud = WordCloud(max_font_size=None, stopwords=stopwords, background_color='white',
                      width=1200, height=1000, collocations=False).generate(text)
plt.figure(figsize=(12, 8))
plt.imshow(wordcloud)
plt.title('Top words')
plt.axis("off")
plt.show()
```

Figura 10 - Nuvem de palavras geradas com todo o texto do conjunto de treinamento



No resultado, a palavra que apareceu com maior frequência foi ‘br’, o que não era esperado por não ser uma palavra da língua inglesa. Analisando o texto, foi possível identificar a existência da *tag*
 no conteúdo dos comentários. Essas

tags foram retiradas de todos os comentários, assim como as palavras *movie* e *film* que são muito genéricas e não geram nenhuma informação. Na sequência, foram geradas novas nuvens de palavras para o conjunto de treinamento (a esquerda) e de teste (a direita).

Nas nuvens de comentários positivos foi verificada uma grande frequência de palavras positivas como well, good, love e funny, o que era esperado. E houve uma similaridade de palavras e frequência entre os conjuntos de treinamento e teste, conforme demonstrado na figura 11.

Figura 11 - Nuvem de palavras para os comentários positivos

Já na nuvem de palavras negativas do conjunto de treinamento (figura 12), apesar de exibidas palavras negativas, como bad, worst, terrible ou awful, foram apontadas algumas ocorrências de palavras positivas, principalmente, as palavras *like* e *good*, assim elas serão analisadas nos próximos passos. Assim como nos comentários positivos, houve também uma boa similaridade entre as palavras e suas frequências entre os conjuntos de treinamento e teste.

Figura 12 - Nuvem de palavras para os comentários negativos

Vale ressaltar que *like* tem dois significados (gostar e como) e pode ser que o último estava impactando o resultado da nuvem de palavras. Desta maneira, foram analisados os 5 primeiros exemplos de comentários negativos que contém a palavra *like*:

- *… once again like it's predecessors i can …*
- *… the story looms like a fart in the room …*
- *… the middle distance: "i don't like who i'm becoming."*
- *… what a "walker" may have been like (say twenty years ago).*
- *… whom i ordinarily like when he's properly cast.*

Ficou evidente que a palavra *like* é usada mais frequentemente no sentido de como, logo os modelos só conseguirão usar a palavra se conseguirem identificar o contexto na qual a mesma é utilizada.

Analisadas as cinco primeiras ocorrências da palavra good em comentários negativos, verificou-se que as expressões estão sendo utilizadas no sentido positivo e isso nos gera um ponto de alerta, assim espera-se que o modelo não entenda que o palavra *good* seja usada como uma característica clara de comentários positivos:

- *… it's better than you might think with some good cinematography by future great vilmos …*
- *… as i reckon this could have been a pretty good film if made properly …*

- *… idea of a woody allen drama unpalatable. and for good reason: they are unbearably wooden …*
- *… it didn't even seem to make sense really. the only good thing about this film was woody …*
- *… one of them being italian! that's kind of a good reason to cast someone else …*

Segue o código utilizado para a extração desses exemplos de *like* e *good*:

Figura 13 - Extração das cinco primeiras ocorrências de like e good em comentários negativos do conjunto de treinamento

```
#%% verificando palavras like e good dos comentários negativos

like_negativos = df_train.loc[(df_train['0ou1'] == 0) & (df_train['text'].str.contains(' like '))]

for i in range(0,5):
    index = like_negativos.iloc[i]['text'].find(' like ')
    print(like_negativos.iloc[i]['text'][index-30:index+30])

good_negativos = df_train.loc[(df_train['0ou1'] == 0) & (df_train['text'].str.contains(' good '))]

for i in range(0,5):
    index = good_negativos.iloc[i]['text'].find(' good ')
    print(good_negativos.iloc[i]['text'][index-50:index+50])
```

4.2 Comparação dos dados estruturados

Foram analisados os dados estruturados dos filmes que foram obtidos por meio das tabelas de dados básicos e de avaliação. Importante lembrar que foi verificado que não existe comentário de um mesmo filme nos conjuntos de treinamento e teste ao mesmo tempo, pois isso facilitaria as predições pela possibilidade de terem expressões próprias de determinado filme. Os scripts que geram os gráficos são mostrados nas figuras 14 e 15, e os gráficos são mostrados nas figuras 16 e 17.

Figura 14 - Script para a geração dos histogramas

```
columns = ['titleType', 'startYear', 'endYear', 'runtimeMinutes', 'averageRatings', 'numVotes', 'sentiment']

for i, col in enumerate(columns):
    plt.hist(df_train[col], alpha=0.5, label='train')
    plt.hist(df_test[col], alpha=0.5, label='test')
    plt.legend(loc='upper left')
    plt.title(col)
    plt.show()
```

Figura 15 - Script para a geração dos gráficos de gênero

```
#%% genres
genres_train = df_train['genres'].str.get_dummies(sep=',')
genres_test = df_test['genres'].str.get_dummies(sep=',')

genres_train['dataset'] = 'train'
genres_test['dataset'] = 'test'

genres = pd.concat([genres_train, genres_test])

fig, axes = plt.subplots(7, 4, figsize=(20, 30))
for ax, col in zip(axes.flat, genres.columns[0:-1]):
    sns_plot = sns.countplot(y = col, hue='dataset', data=genres, ax=ax)
```

Figura 16 - Gráficos das informações básicas e de avaliação

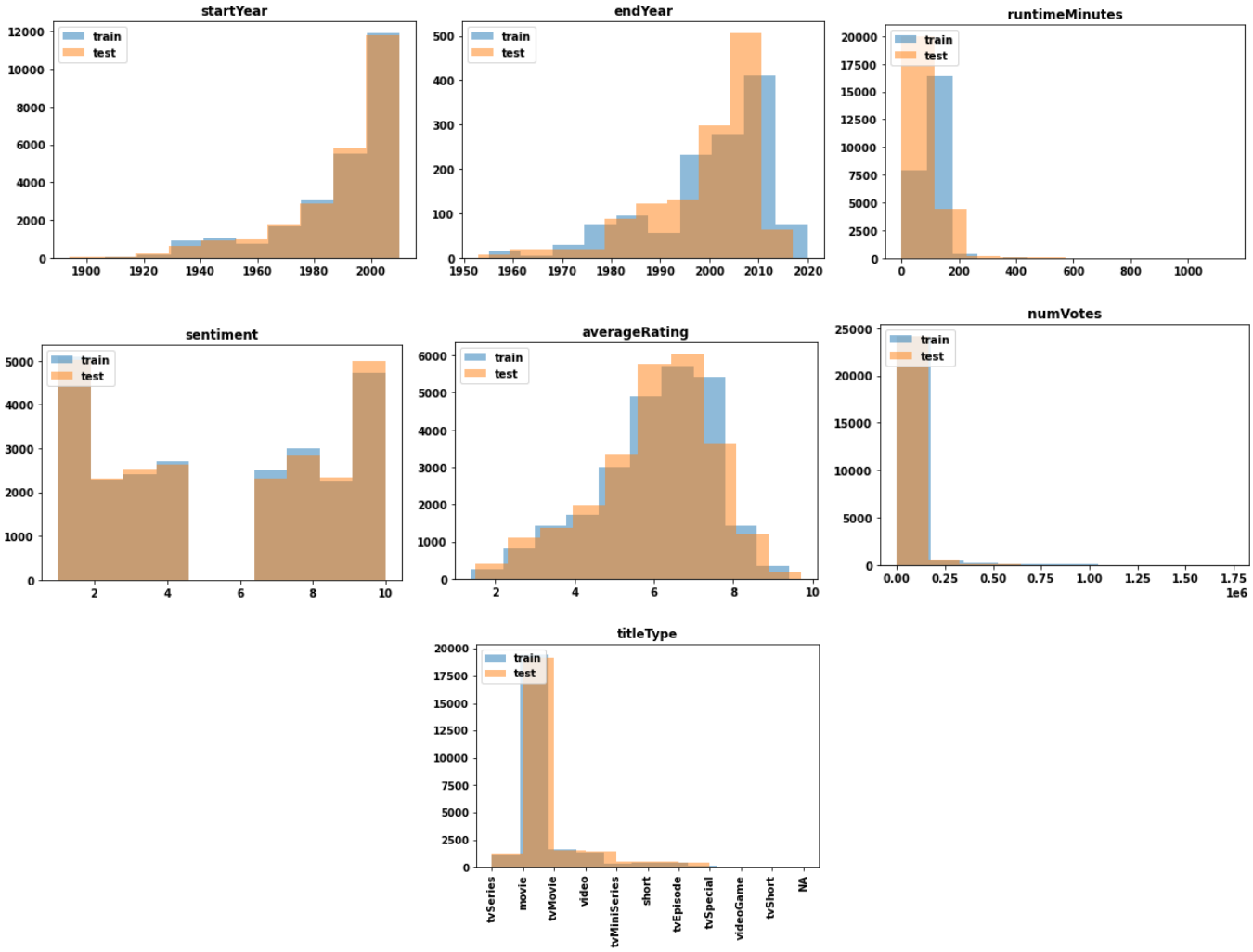
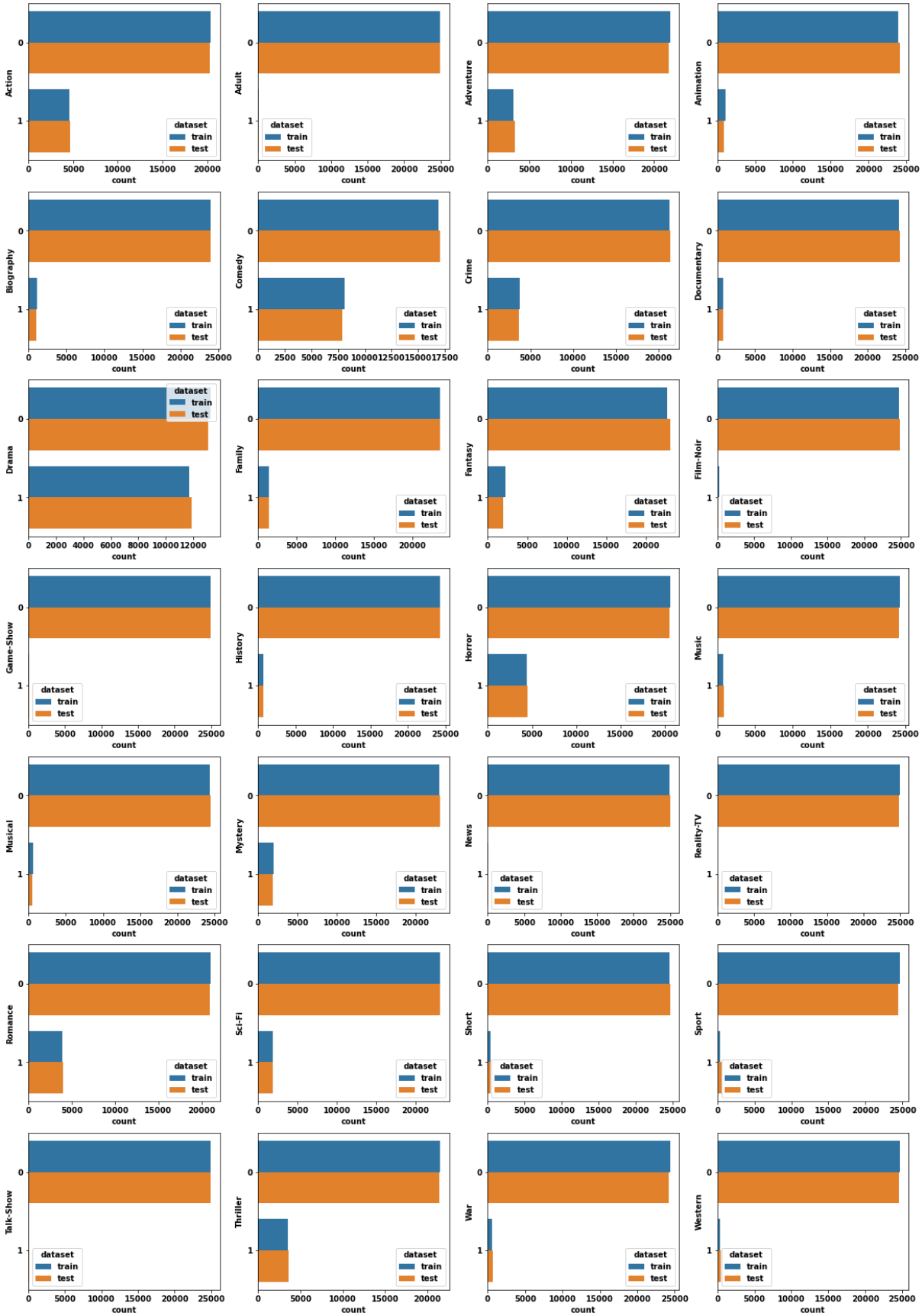


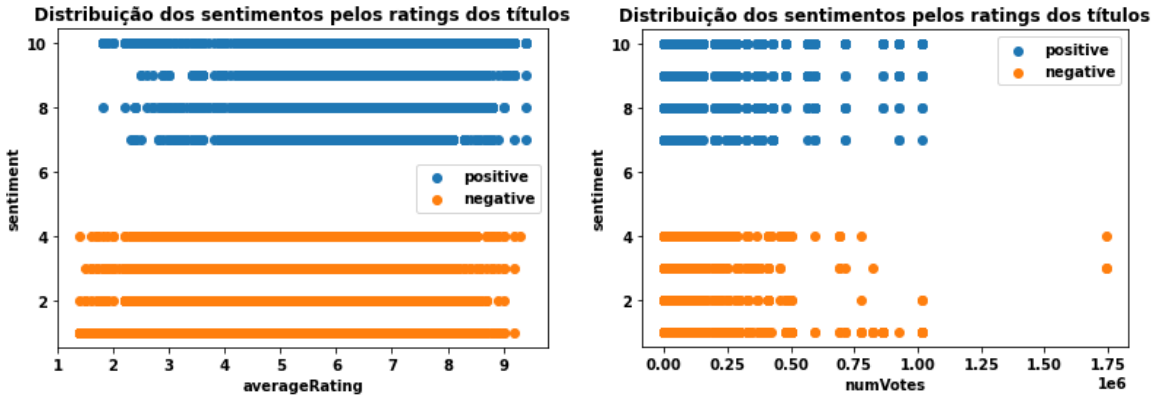
Figura 17 - Gráficos com a comparação de gênero



Conclui-se que os dados de treinamento e teste possuem a mesma distribuição estatística em todas as categorias levantadas, mostrando a qualidade do dataset que está sendo utilizado neste trabalho.

No gráfico 18 é mostrada a distribuição entre o *sentiment* e o *averageRatings* ou *numVotes*, mostrando que temos uma boa distribuição de comentários positivos e negativos em títulos bem ou mal avaliados e muito ou pouco votados. Desta maneira, seria possível utilizar essas colunas como *features* já que não se teria uma informação privilegiada, facilitando as predições dos comentários. Apesar disso, decidiu-se por não os utilizar, pois não seria possível obter os valores de averageRatings e numVotes de títulos recentes, impactando a predição de sentimento.

Figura 18 - Gráficos comparando o sentiment com avarageRating e numVotes



5 Criação de Modelos de Machine Learning

5.1 *Bag of words*

Esse modelo usa cada palavra dos comentários como *features*, ou seja, para cada comentário será gerado um vetor no qual cada eixo é uma palavra. Os eixos são todas as palavras encontradas no conjunto de treinamento, exceto pelas *stopwords (palavras consideradas menos relevantes como artigo, conjunção e pontuação)*. Para gerar esses vetores foram testadas duas funções do *sklearn*, *CountVectorizer* e *TfidfVectorizer*, com *ngrams* (conjunto de palavras) igual a 1 ou 2. No *CountVectorizer* cada eixo terá a quantidade das palavras contadas para determinado comentário. Já o *TfidfVectorizer* faz uma ponderação para dar menos peso para palavras que são

encontradas frequentemente no *dataset,* por exemplo, a palavra one que é uma das mais frequentes (conforme demonstrado na nuvem de palavras), terá um valor baixo, pois entende-se que ela não gera muita informação.

Figura 19 - Funções CountVectorizer e TfidfVectorizer do Sklearn

```
if tfidf:
    vectorizer = TfidfVectorizer(stop_words=stopwords, ngram_range=(1, ngram))
else:
    vectorizer = CountVectorizer(stop_words=stopwords, ngram_range=(1, ngram))
```

Para este modelo também foi testado incluir as *features* da base de dados básicos. Realizou-se também a stemização, que é o processo de reduzir palavras flexionadas a sua palavra raiz, por exemplo, *waiting, waited, wait* e *waits* foram todos transformados em *wait*. Para a realização dessa técnica foi utilizada a função *PorterStemmer* do pacote *nltk*.

Figura 20 - Função para stemizar um comentário

```
def stem_sentences(sentence):
    stemmer = nltk.stem.PorterStemmer()
    tokens = sentence.split()
    stemmed_tokens = [stemmer.stem(token) for token in tokens]
    return ' '.join(stemmed_tokens)
```

Outra técnica que foi aplicada foi a lematização (fugura 21), parecida com a técnica anterior, porém faz uma análise morfológica nas palavras para pegar o radical ao invés de cortar prefixos ou sufixos no começo ou final da palavra, como faz o stemização. Para a realização dessa técnica foi utilizado o pacote *spacy*. Aproveitando o pacote, incluiram-se também outras *features* como *pos-tagging* (extrai a categoria morfo-sintática das palavras, como verbo, substantivo, adjetivos, entre outros), reconhecimento de entidades (extrai a entidade das palavras, como organização, numeral, pessoa, entre outros) e vetor normalizado (normalização euclidiana dos vetores de cada palavra). Para os dados de *pos-tagging* e reconhecimento de entidades utilizou-se o mesmo método usado com as palavras, ou seja, *CountVectorizer* ou *TfidfVectorizer*.

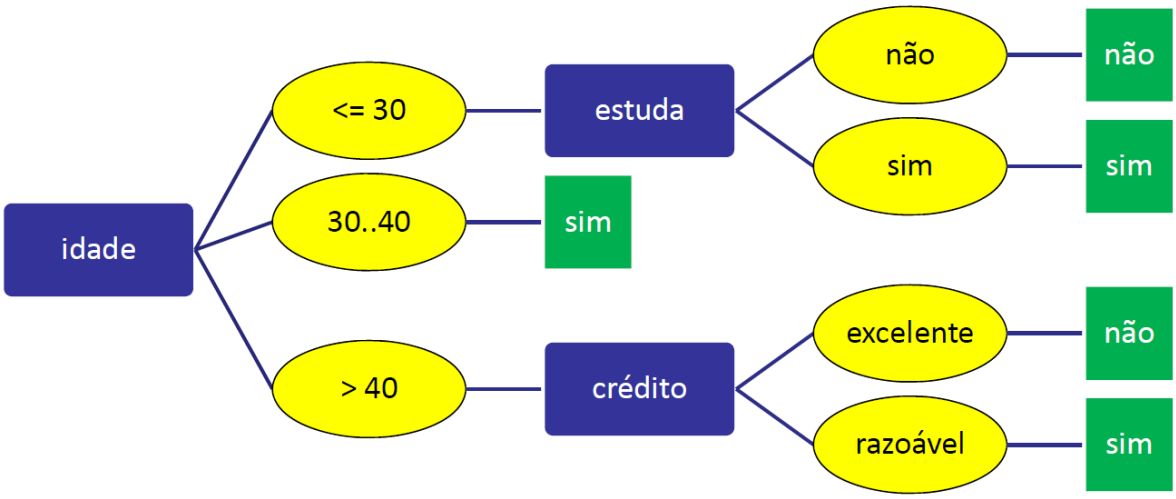
Figura 21 - Classe que realiza a lematização e extração de features

```
class Lemmatization(object):

    def __init__(self, model):
        self.nlp = spacy.load(model)

    def lemma_sentences(self, sentence):

        pos = ""
        lemma = ""
        ent_type = ""
        vector_norm = ""
        # tag = ""
        # cluster = ""
        for token in self.nlp(sentence):
            # text += token.text + " "
            pos += token.pos_ + " "
            lemma += token.lemma_ + " "
            ent_type += token.ent_type_ + " "
            vector_norm += str(token.vector_norm) + " "
            # tag += token.tag_ + " "
            # cluster += str(token.cluster) + " "
        return pd.Series([lemma, pos, ent_type, vector_norm])
```

Para a realização do treinamento foi escolhido o pacote LightGBM (https://lightgbm.readthedocs.io/en/latest/), que é um framework de gradient boosting baseado em árvores de decisão, extremamente eficiente por utilizar pouca memória, ter rápido processamento e ser um modelo com excelente acurácia se comparado a outros frameworks disponíveis.

Árvore de decisão é um método que cria regras simples para cada nó seguindo uma sequência até chegar no nó folha que contém a resposta para o problema (como o exemplo abaixo).

Figura 22 - Exemplo de uma árvore de decisão



Gradient Boosting é uma técnica de ensemble que foca nos erros de um modelo gerado anteriormente, melhorando a precisão do algoritmo de aprendizado. Para cada nova árvore de decisão que o LightGBM gera, ele usa a técnica citada para tentar melhorar a acurácia do modelo. Na figura 23, apresenta-se o código para treinar o modelo e realizar as predições do conjunto de teste.

Figura 23 - Código de treinamento utilizando o lightGBM

```
params = {}
params['boosting_type'] = 'gbdt'
params['objective'] = 'binary'
params['metrics'] = 'binary_error'
params['learning_rate'] = 0.1
params['num_leaves'] = 20
params['min_data_in_leaf'] = 30
params['lambda_l2'] = 0.1

d_train_cv = lgb.Dataset(x_train, label=y_train, feature_name=vocab)
#Cross validation to get best iteration
bst = lgb.cv(params,
         d_train_cv,
         nfold = CV,
         shuffle = True,
         stratified = True,
         num_boost_round=10000,
         early_stopping_rounds=100,
         verbose_eval=50,
         seed = 70)
#get best iteraton
cv_binary_loss = min(bst.get('binary_error-mean'))
best_iter = bst.get('binary_error-mean').index(cv_binary_loss)
#train
m = lgb.train(params, train_set=d_train_cv, num_boost_round=best_iter, verbose_eval=best_iter)
preds_train = m.predict(x_train)
#find optimal cutoff
threshold = Find_Optimal_Cutoff(y_train, preds_train)[0]
#predict
preds_test = m.predict(x_test)
test_acc, test_roc, test_f1 = score(y_test, preds_test, threshold)
#save results
results_train = pd.read_csv('./results/results_train.csv', index_col=0)
results_train = results_train.append({'params':params, 'tfidf':tfidf, 'cv':CV,
                          'feat_from_dataset': feat_from_dataset,'feat_from_lemma': feat_from_lemma,
                          'StemOrLemma': StemOrLemma, 'cv_binary_loss':cv_binary_loss,
                          'test_acc':test_acc, 'test_roc':test_roc, 'test_f1':test_f1}, ignore_index=True)
results_train.to_csv('./results/results_train.csv')
```

5.2 Word embeddings

Podemos definir *word embedding* como sendo um conjunto de técnicas que mapeia a semântica e sintática de uma linguagem natural em um espaço real utilizando estatísticas. Dessa forma, palavras de um conjunto de texto são mapeadas para vetores reais.

Assim palavras como cachorro e gato são mapeadas em vetores próximos devido ao seu grau de similaridade. E é possível fazer operação com esses vetores, por exemplo, se fosse fazer a conta (rainha – mulher) + homem, o vetor retornado será próximo de rei. Isso é importante pois como os comentários são conjunto de palavras, realizamos a soma e a média das palavras para gerar o embedding do comentário.

Como cada eixo do vetor representa uma característica das palavras e palavras similares são próximas, foi utilizar o *word embeddings* para gerar as nossas *features* numa dimensão bem menor que no método anterior. Para efeito de comparação, foi gerado vetor de 1860006 dimensões no modelo de bag of words enquanto que com o *word embeddings* os vetores utilizados foram de 300 dimensões.

Nesta etapa, foram utilizados modelos pré-treinados com diferentes métodos e diferentes corpus, foram eles:

- Word2vec – modelo *enwiki_20180420 treinado com dados da Wikipédia:* https://wikipedia2vec.github.io/wikipedia2vec/pretrained/
- Glove – modelo glove.42B.300d.zip treinado com dados do Common Crawl: https://nlp.stanford.edu/projects/glove/
- FastText – modelo crawl-300d-2M.vec.zip treinado com dados Common Crawler: https://fasttext.cc/docs/en/english-vectors.html
- Spacy – modelo en_core_web_md treinado com dados Commo Crawler e OntoNotes 5: https://spacy.io/models/en

Como no método anterior, utilizamos o LightGBM como modelo para treinamento e predição.

Figura 24 - Classe que extrai o vetor de uma palavra e realiza a operação de soma ou média para um determinado texto

```
class CalcEmbeddingVectorizer:

    def __init__(self, word_model, serie, glove=False):
        if not glove:
            self.embeddings = KeyedVectors.load_word2vec_format(word_model, binary=False, unicode_errors="ignore")

        else:
            tmp_file = get_tmpfile("test_word2vec.txt")
            _ = glove2word2vec(word_model, tmp_file)
            self.embeddings = KeyedVectors.load_word2vec_format(tmp_file)

    def word_average(self, sent):
        mean=[]
        for word in sent:
            try:
                mean.append(self.embeddings.get_vector(word))
            except:
                continue
        if not mean:
            return np.zeros(self.embeddings.vector_size)
        else:
            mean = np.array(mean).mean(axis=0)
            return mean

    def word_sum(self, sent):
        sum_=[]
        for word in sent:
            try:
                sum_.append(self.embeddings.get_vector(word))
            except:
                continue
        if not sum_:
            return np.zeros(self.embeddings.vector_size)
        else:
            sum_ = np.array(sum_).sum(axis=0)
            return sum_
```

5.3 Transfer learning

Por último, utilizou-se a técnica de *transfer learning* que consiste em pegar um modelo pré-treinado e ajustá-lo com os textos de um determinado projeto. Os modelos de embeddings anteriores não foram treinados especificamente para texto de comentários de usuários e fazer esse ajuste nos trará vetores mais ajustados a realidade do problema proposto. Nesta fase foi realizada em nuvem por exigir um maior poder de processamento e foi escolhido o Kaggle por ser a plataforma mais simples de utilizar e com 30 horas gratuitas por semana de processamento em GPU.

Decidiu-se utilizar a biblioteca fastai (https://www.fast.ai/) que possui classes prontas para realizar tanto o transfer learning quanto a classificação dos comentários. Os modelos utilizados foram Redes Neuronais Recorrentes (RNN) que levam em consideração a sequência de palavras escritas e logo o contexto delas dentro de uma frase ou parágrafo.

Para este trabalho foi necessário criar um conjunto com todos os comentários do conjunto de treinamento, inclusive os não anotados, para criarmos um corpus maior para ajustar o modelo aos textos de comentários do IMDb.

De posse desses dados, foi realizado um pré-processamento para criar um conjunto de dados ajustado para o treinamento de rede neural. Assim todos os textos são concatenados e o alvo é a próxima palavra da frase, além disso é colocada diversas *tags* que são utilizadas pelos modelos (*tags* para marcar o início de um comentário, início de uma frase, palavras com letras maiúsculas, entre outros).

Figura 25 - Fazendo o pré-processamento dos comentários

```
        data_lm = TextLMDataBunch.from_csv(path, 'df_fastai.csv', text_cols='text', label_
cols='label')
```

Feito esse passo, foi importado um modelo pré-treinado e realizado o aprendizado com os textos dos comentários do IMDb.

Figura 26 - Importação de um modelo pré-treinado

```
learn_lm = language_model_learner(data_lm, AWD_LSTM, drop_mult=0.3)
```

Figura 27 - Realizando o ajuste da rede neural para os dados dos comentários do IMDb

```
learn_lm.fit_one_cycle(1, lr*10, moms=(0.8,0.7))
```

```
learn_lm.unfreeze()
```

```
learn_lm.fit_one_cycle(5, lr, moms=(0.8,0.7))
```

O modelo com os embeddings ajustados para os comentários do conjunto de treinamento e não anotados foi gerado e o próximo passo foi realizar o treinamento da RNN, mudando o alvo para a classificação do sentimento dos comentários conforme figura 28.

Figura 28 - Treinamento da RNN para a classificação dos comentários utilizando os encoders gerados no passo anterior

```
data_clas = TextClasDataBunch.from_csv(path, 'text_train_labeled.csv', test='text_test.csv', vocab=data_lm.train_ds.vocab, bs=32)
```

```
learn_c = text_classifier_learner(data_clas, AWD_LSTM, drop_mult=0.3)
```

```
learn_c.load_encoder('fine_tuned_enc_2')
learn_c.freeze()
```

```
learn_c.fit_one_cycle(1, 1e-2, moms=(0.8,0.7))
```

```
learn_c.freeze_to(-2)
learn_c.fit_one_cycle(1, slice(1e-2/(2.6**4),1e-2), moms=(0.8,0.7))
```

```
learn_c.freeze_to(-3)
learn_c.fit_one_cycle(1, slice(5e-3/(2.6**4),5e-3), moms=(0.8,0.7))
```

```
learn_c.unfreeze()
learn_c.fit_one_cycle(3, slice(1e-3/(2.6**4),1e-3), moms=(0.8,0.7))
```

6 Apresentação dos Resultados

6.1 Bag of words

Como descrito na etapa anterior, realizaram-se vários testes para ver qual seria o melhor modelo de *bag of words*, o melhor modelo foi utilizando o *CountVectorizer*, sem utilizar as *features* dos dados básicos, sem utilizar stemização nem lematização, e utilizando as *features* de *pos-tagging*, reconhecimento de entidades e vetor normalizado e, por fim, usando *ngrams* igual a 2.

Tivemos os seguintes resultados para esse método:

- Log Loss da Validação Cruzada: 0.11468
- AUC de Teste: 0.96
- F1 Score de Teste: 0.895
- Acurácia de Teste: 0.896

Destaca-se que os outros modelos com configurações diferentes não tiveram uma performance muito pior que a escolhida. Por exemplo, a pior configuração teve acuraria de 0.875 e AUC de 0.946 no conjunto de teste.

A curva ROC e a matrix de confusão da predição do modelo escolhido para o conjunto de teste são apresentados abaixo:

Figura 29 - Curva ROC do conjunto de teste

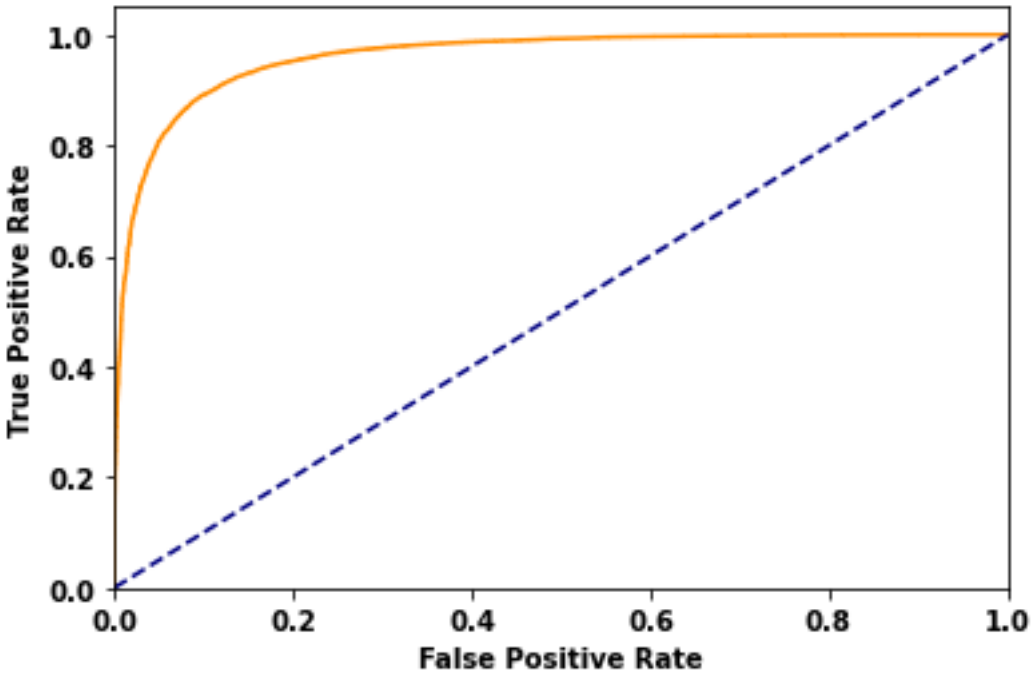
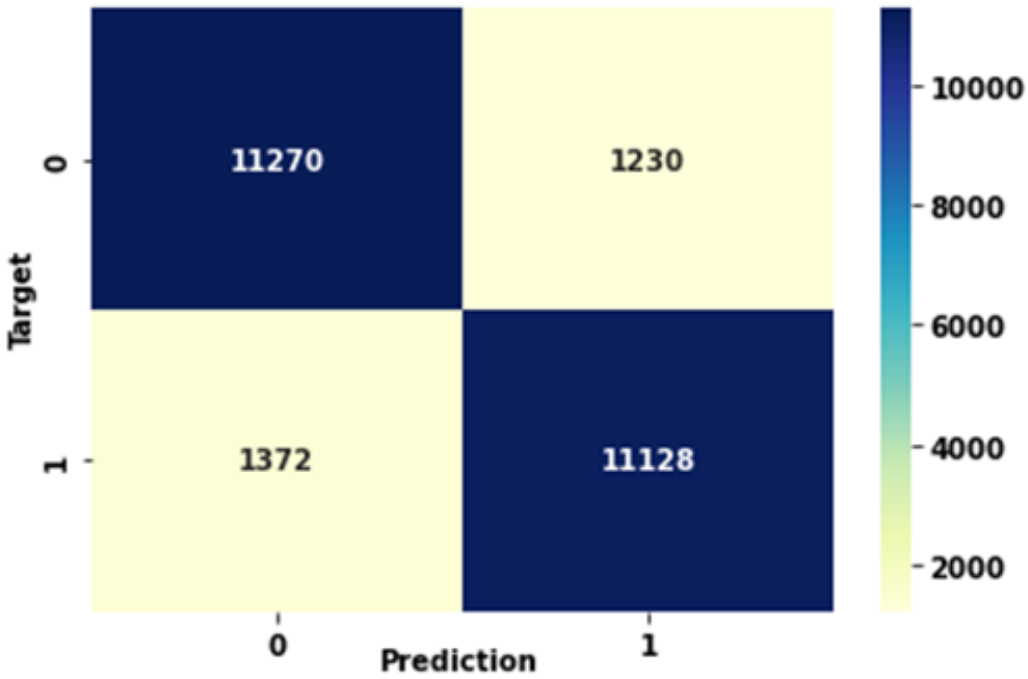


Figura 30 - Matriz de confusão para o conjunto de teste



A explicabilidade é a grande vantagem desse modelo, pois ele permite identificar quais são as palavras mais importantes para o modelo (figura 31) e para cada

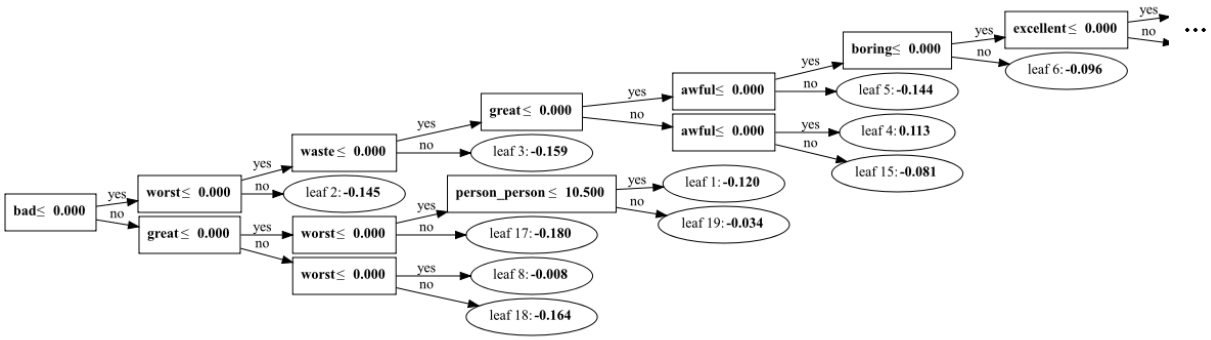
comentário específico. As principais *features* têm significado negativo corroborando o achado de palavras com sentido negativo em comentários positivos. Além disso, a palavra *good* não aparece pois como ela é mencionada tanto em avaliações negativas como positivas, ou seja, ela não foi considerada muito relevante para o modelo, que atribuiu importância de 286, significativamente baixa quando comparada com a de seu antônimo *bad (9463)*.

Figura 31 - Importância das features do modelo

features	feat_import_split	feat_import_gain
bad	43	9462.6945
worst	40	7481.0223
waste	23	4280.9093
great	42	4111.0808
awful	34	3143.2203
excellent	33	2399.3335
boring	30	1927.1389
best	37	1707.0427
wonderful	28	1651.5365
terrible	21	1558.5014
nothing	31	1517.7727
poor	28	1446.6917
perfect	32	1444.5841
stupid	15	1239.1402
poorly	30	1153.2958
worse	23	1140.6132
well	24	1062.0364
horrible	34	1040.7654
dull	27	976.3129
amazing	19	938.4137
love	20	897.1846

É possível visualizar todas as árvores (veja uma parte da primeira árvore do modelo na figura abaixo), mas é humanamente impossível ficar analisando as árvores para cada comentário pois temos 796 árvores, assim serão mostradas as 5 palavras mais relevantes tanto para o comentário ser considerado positivo quanto negativo para avaliar cada um.

Figura 32 – Parte da primeira árvore de decisão do modelo



Segue alguns exemplos:

Tabela 3 - Resultado correto do modelo para um comentário

Comentário	*bill paxton has taken the true story of the 1913 us golf open and made a film that is about much more than an extra-ordinary game of golf. the film also deals directly with the class tensions of the early twentieth century and touches upon the profound anti-catholic prejudices of both the british and american establishments. but at heart the film is about that perennial favourite of triumph against the odds.the acting is exemplary throughout. stephen dillane is excellent as usual, but the revelation of the movie is shia laboeuf who delivers a disciplined, dignified and highly sympathetic performance as a working class franco-irish kid fighting his way through the prejudices of the new england wasp establishment. for those who are only familiar with his slap-stick performances in "even stevens" this demonstration of his maturity is a delightful surprise. and josh flitter as the ten year old caddy threatens to steal every scene in which he appears.a old fashioned movie in the best sense of the word: fine acting, clear directing and a great story that grips to the end - the final scene an*

	affectionate nod to casablanca is just one of the many pleasures that fill a great movie.
Previsão	0.9660271831004925
Target	1
As 5 features mais positivas	• excellent – peso: 0.7471030920567032 • delightful – peso: 0.6923975195494518 • great – peso: 0.5441927983507173 • paxton – peso: 0.40278921406984924 • surprise – peso: 0.29291030898018416
As 5 features mais negativas	• catholic – peso: -0.3548290241314693 • appears – peso: -0.282244854069731 • aux aux – peso: -0.26378270549499166 • fill – peso: -0.23451218232666488 • year old – peso: -0.21193359766472383

Tabela 4 - Resultado incorreto do modelo para um comentário

Comentário	*the basic genre is a thriller intercut with an uncomfortable menage-a-trois. fellowes has tried to make a lot more out of this, using the lies of the title in order to bring about all manner of small twists, invariably designed to surprise the characters more than the audience.it's really rather messy though. fellowes doesn't seem interested presenting the thriller elements in a fashion that will keep us seat-edged. rather his focus is on the moral predicaments themselves.the dialogue is inconsistent, stagey here, vernacular there and with the constant surprise of realism undone by the occasional cliché-landmine. though there is no fussing over the locations so that the actors can get on with existing in their space the dreadful score can't create a further dimension and often works against the emotional momentum of given set pieces. there's also a very prosaic, dare i say it british feel to the filming. i didn't want to see a document of two successful middle class people caught in an extraordinary situation, i wanted to see some sort of artful recounting of the story.finally it is, in fact, the story which lets the rest down. just as the elements of suspense are rather flat so the story is an asymmetric sum of subplots of different shapes and sizes, woven as a vehicle for character examination. wilkinson and watson support this meta-essay with good perfor-*

	mances and john warnaby's ebullient colleague simon to wilkinson is a welcome foil for much of the brow-furrowing.i'm disappointed; not that it's bad, but that it could have been much better. 3/10
Previsão	0.2263536755290465
Target	1
As 5 features mais positivas	• *extraordinary* – peso: 0.39405626834044477 • *watson* – peso: 0.3136195778604735 • *surprise* – peso: 0.26364222629181505 • *caught* – peso: 0.1928502314624285 • *often* – peso: 0.15584095852811505
As 5 features mais negativas	• *dreadful* – peso: -1.0948841125107427 • *bad* – peso: -0.597534419900074 • *disappointed* – peso: -0.2413515478186631 • *clich_* – peso: -0.21273294738870435 • *flat* – peso: -0.20298774832784036

O modelo acertou no primeiro exemplo e as palavras *excelent, delightful* e *great* foram as principais responsáveis para o comentário ser considerado como positivo. Já no segundo exemplo, o modelo erra, pois é enganado pelas palavras *dreadful*, *bad* e *disappointed*. Porém, ao ler o comentário, é possível concordar com o modelo, indicando um provável erro de anotação.

Foi verificada a estatística daqueles que o modelo apresenta maior certeza, ou seja, no qual os resultados são acima de 90% de certeza de serem avaliação positivas ou negativas. Nestes casos, o modelo atingiu 97,6% de acurácia para os positivos e 97.8% de acuraria para os negativos. Os histogramas abaixo mostram esses resultados.

Figura 33 - Histograma com o resultados com predições acima de 0,9

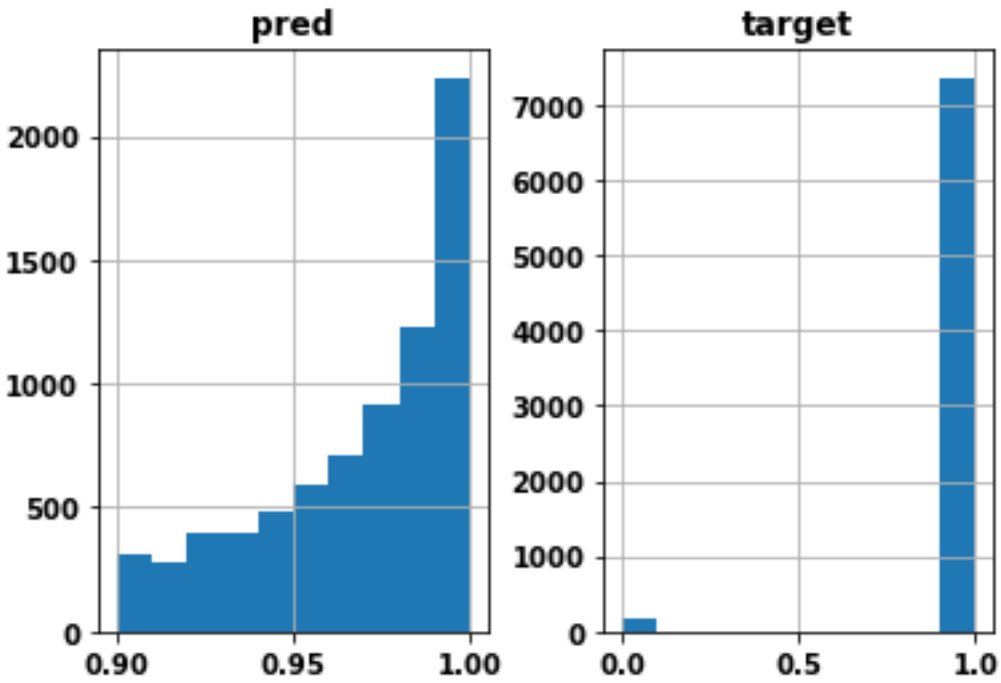
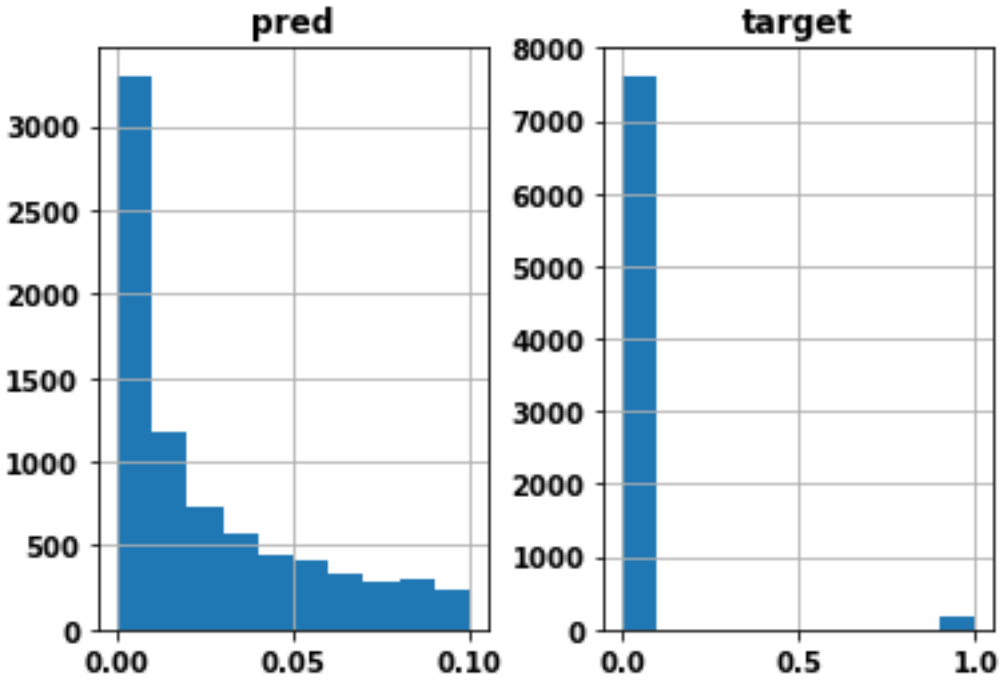


Figura 34 - Histograma com o resultados com predições abaixo de 0,1



Abaixo segue a análise de dois erros, um positivo e um negativo, que o modelo tem muita certeza:

Tabela 5 - Resultado errado do modelo para um comentário negativo

Comentário	*at the bottom end of the apocalypse movie scale is this piece of pish called 'the final executioner'.. at least where i come from. a bloke is trained by an ex-cop to seek vengeance on those that killed his woman and friends in cold blood.. and that's about it. lots of fake explosions and repetitive shootings ensue. has one of the weirdest array of costumes i've seen in a film for a while, and a massive fortress which is apparently only run by 7 people. great job on the dubbing too guys(!) best moment: when our hero loses a swordfight and is about to be skewered through the neck, he just gets out his gun and bang! why not do that earlier? it's a mystery. as is why anyone would want to sit through this in the first place. i'm still puzzling over that one myself now.. 2/10*
Previsão	0.955907039508908
Target	0
As 5 features mais positivas	• *vengeance* – peso: 0.7424489910950656 • *great job* – peso: 0.6181543870458498 • *trained* – peso: 0.30719313400117704 • *great* – peso: 0.2886228036754609 • *best* – peso: 0.28560263143071035
As 5 features mais negativas	• *fake* – peso: -0.22103667767053495 • *least* – peso: -0.14067169590123335 • *cconj* – peso: -0.13964564134780263 • *gun* – peso: -0.1228206791064163 • *woman* – peso: -0.10499288682615651

Claramente o modelo foi enganado por palavras positivas (*great job* e *best*). Além disso, a palavra *vengeance* não tem um sentido positivo, mas como o seu peso é muito grande entende-se que há muitas avaliações positivas com *vengeance* no conjunto de treinamento.

Tabela 6 - Resultado errado do modelo para um comentário positivo

Comentário	*madonna gets into action, again and she fails again! who's that girl was released just one year after the huge flop of shangai surprise and two after the successful cult movie desperately seeking*

	susan. she chose to act in it to forget the flop of the previous movie, not suspecting that this latter could be a flop, too. the movie received a bad acceptance by american critic and audience, while in europe it was a success. madonna states that "some people don't want that she's successful both as a pop star and a movie-star". the soundtrack album, in which she sings four tracks sells well and the title-track single was agreat hit all over the world, as like as the world tour. the truth isthat madonna failed as an actress 'cause the script was quite weak. butit's not so bad, especially for those who like the 80's: it's such a ramshackle, trash, colorful and joyful action movie ! at the end, it's very funny to watch it.
Previsão	0.04250326698008927
Target	1
As 5 features mais positivas	• *surprise* – peso: 0.23916602526288402 • *world* – peso: 0.21516665044277372 • *especially* – peso: 0.18060202406236067 • *soundtrack* – peso: 0.17971680078799138 • *cconj* – peso: 0.17943073279865177
As 5 features mais negativas	• *fails* – peso: -1.0654592976242014 • *bad* – peso: -1.0169933855393445 • *weak* – peso: -0.45334595457774585 • *flop* – peso: -0.3935453764520817 • *failed* – peso: -0.3760558133831129

Assim como no exemplo anterior, observa-se que o modelo é enganado com várias palavras negativas como *fails, bad, weak* e *failed*. Porém, ao ler o comentário, é possível ter a impressão de ser uma avaliação negativa a não ser pela última frase do texto.

6.2 Word embeddings

Dos 4 modelos pré-treinados utilizados, o que obteve o melhor resultado foi realizando a média dos embeddings do modelo FastText. Assim, segue o resultado dele.

- Log Loss da Validação Cruzada: 0.145

- AUC de Teste: 0.932
- F1 Score de Teste: 0.853
- Acurácia de Teste: 0.854

Para esse método a diferença de performance em relação as outras configurações de *word embedding* não foram relevantes, a maioria teve performance em torno de 0.83 de acurácia e 0.915 de AUC no conjunto de teste.

Este modelo escolhido teve um score pior na validação cruzada e também no conjunto de teste, em relação ao modelo anterior, houve uma diferença de aproximadamente 4% de acurácia para o conjunto de teste.

Abaixo a curva ROC e a matriz de confusão das previsões do modelo para o conjunto de teste:

Figura 35 - Curva ROC para o conjunto de teste

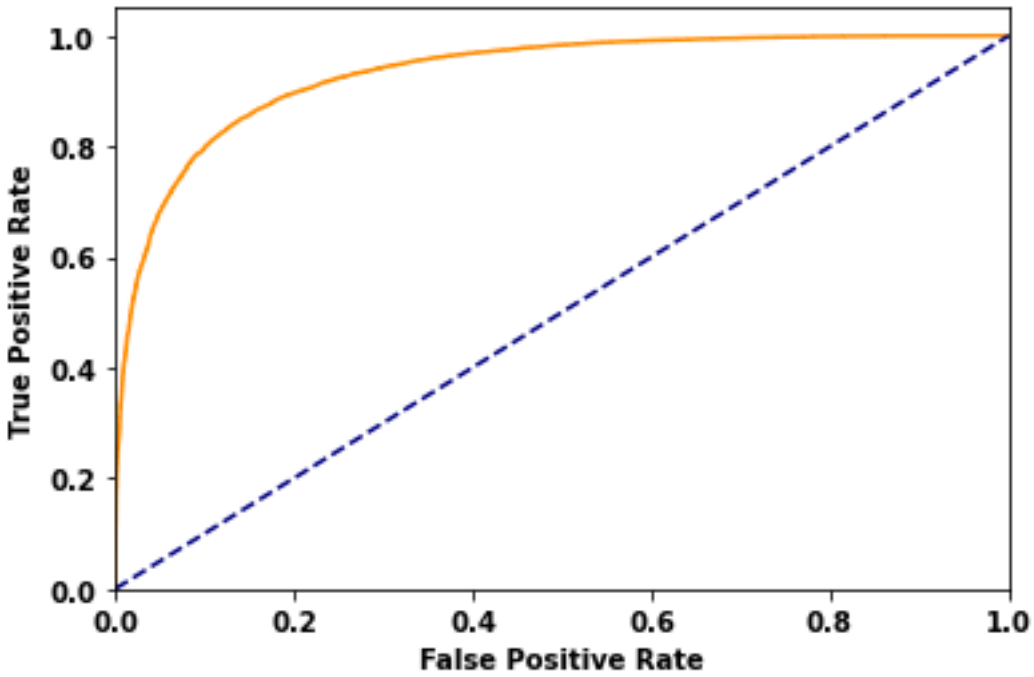
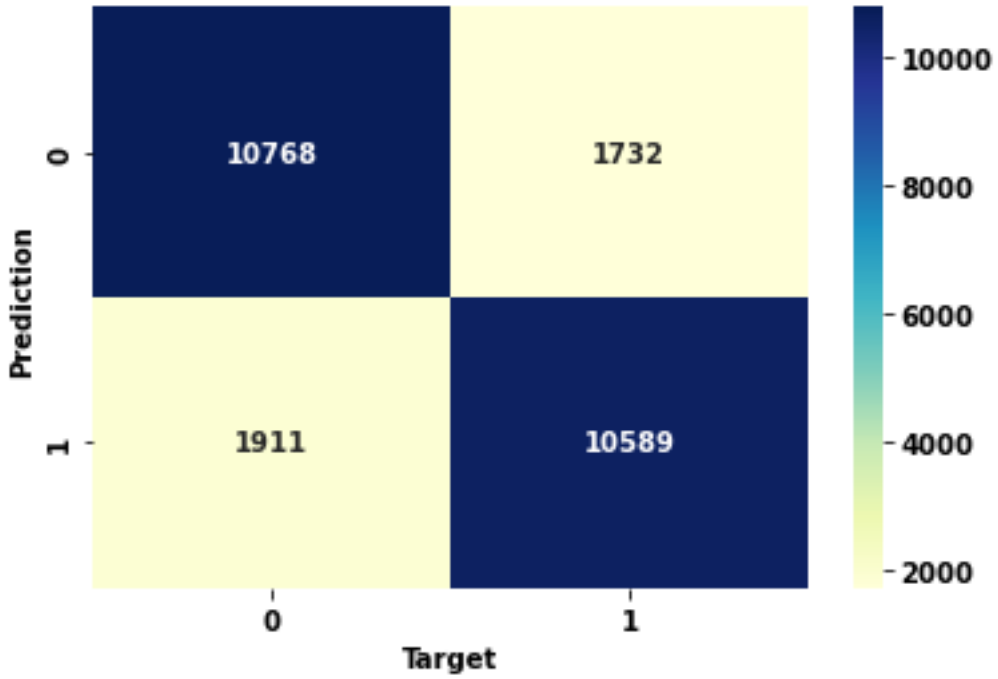


Figura 36 - Matriz de confusão para o conjunto de teste



Neste modelo não é possível mostrar quais são as *features* mais importantes, pois cada feature é um eixo no vetor do *embedding* e não é humanamente compreensível.

Segue o desempenho do modelo nos comentários que tem mais certeza:

Figura 37 - Histograma com o resultados com previsão maior que 0,9

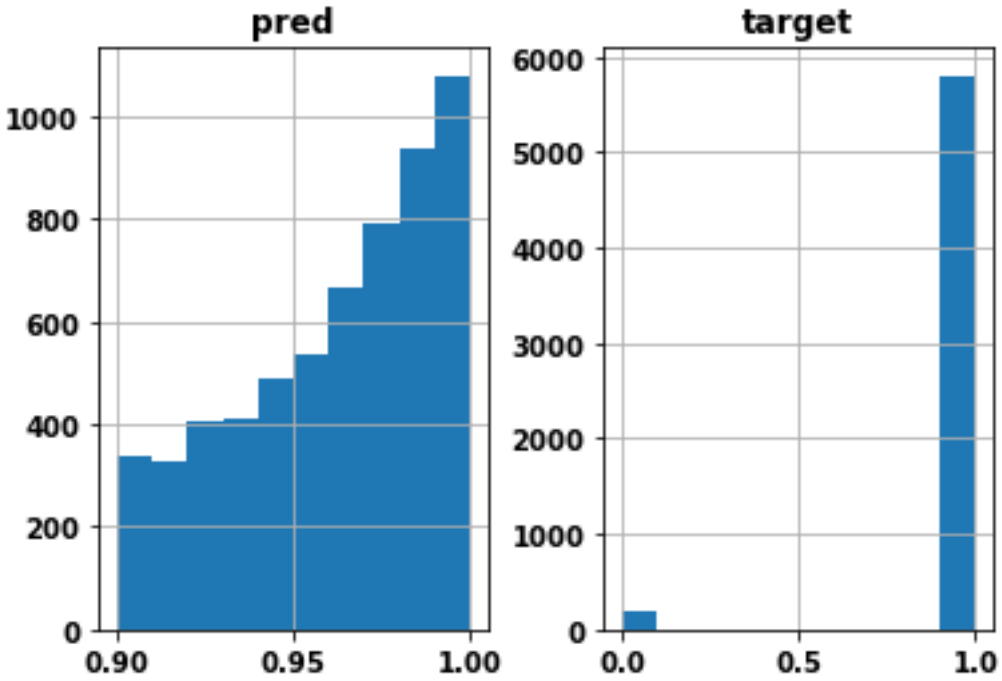
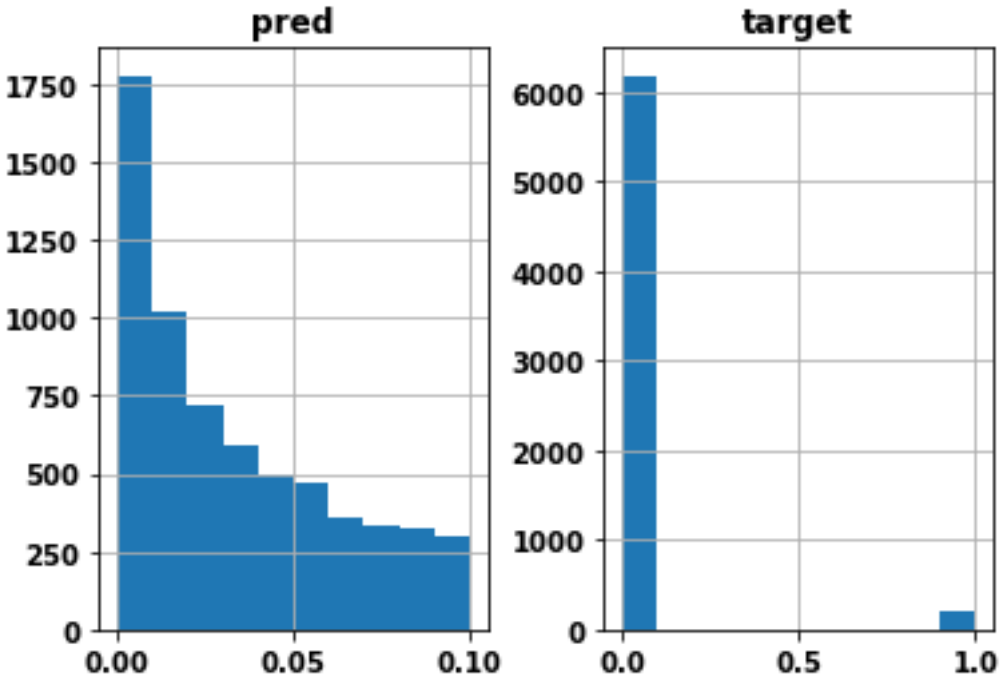


Figura 38 - Histograma com o resultados com previsão menor que 0,1



Foi obtido um resultado de acurácia de 96,89% para os comentários positivos e de 96,84% para comentários negativos. Conforme foi feito no modelo anterior, segue um exemplo positivo e outro negativo que o modelo errou tendo grande confiança:

Tabela 7 - Resultado errado do modelo para um comentário positivo

Comentário	*disappointed documentaryi thought would second chess match grandmaster garry kasporov deep blue supercomputer designed ibm computer experts beat human chess playerkasparov still is-considered greatest chess player everthe movie takes us back 1997 kasporov agreed rematch deep blue defeating 1 year earli-erbut instead focusing gameit focuses happens afterthere snip-pets gamebut manymuch film centers around kasporovs paranoid obsession match rigged part conspiracy theory lost match unfair-lythe movie also includes interviews people interesting waythey even chat manager building match took placewho caresi also found dry slowultimately movie unsatisfyingthis opinionof courseif like conspiracy theoriesthis movie might interest youfor people chess conspiracy theoriesthis movie would probably valuei chess fanand stuck thati givegame overkasparov machine 410*
Predição	0.927
Target	0

Tabela 8 - Resultado errado do modelo para um comentário negativo

Comentário	*madonna isnt meryl streep still one first films comedy give break sure movie mediocre best pales comparison earlier counterpart w katherine hepburn bringing baby thougha piece fluff quite bit fun watch ive yet hear anyone slams madonnas acting skills back w evidence even adjectives awful bad vague descriptive words wanna see bad acting justify argument singers stick singing whit-ney houston shes undeserved commercial success actress his-tory couldnt act way hatbox american public obviously cannot dis-cern difference credible performance movie star power think ma-donna always least credible movies get real people madon-nabashing 90s*
Predição	0. 01124
Target	1

Target	1

Nestes casos não temos como explicar o porquê dos erros.

6.3 Transfer learning

Primeiramente, apresenta-se o resultado do modelo de linguagem. Seguem dois exemplos de preenchimento pelo modelo para cada início de frase:

- *Sentença "The film is" com 30 palavras:*
  - *The film is an allegory on the American society , and the way the American people have dealt with it . This film is a study of how the*
  - *The film is a total drag . The plot is thin , the acting is bad (a great example of what a poor director can do without starring the other way*

- Sentença "*I liked this film because*" com 40 palavras:
  - *I liked this film because it was a true story , but it seemed to be so victim of the events that the Have Committed War played out in the South . The Vietnam War , which was*
  - *I liked this film because it was made with the only sex and nudity that has been used in movies . It is , on the whole , a very enjoyable film , and the movie is hardly boring . The action is*

- Sentença "I would not say this movie" com 40 palavras:
  - *I would not say this movie is good . It is just another story that is written by someone who has not seen it yet . This is a crude movie that pokes fun at the industry . As you do n't have*
  - *I would not say this movie is as bad as it is in imdb , but in that case , it is by far the worst movie i have ever seen . This movie is about a bunch of people searching for a missing brother*

O modelo conseguiu se ajustar muito bem aos comentários do IMDb pois o estilo de texto gerado é bem parecido com o comentário de um humano. O modelo teve uma acurácia de 30% para a predição da próxima palavra.

Depois de analisar o modelo da linguagem, segue o resultado do modelo de classificação e logo depois a curva ROC e a matriz de confusão:

- Log Loss da Validação Cruzada: 0.1746
- AUC de Teste: 0.982
- F1 Score de Teste: 0.934

- Acurácia de Teste: 0.936

Figura 39 - Curva ROC para o conjunto de teste

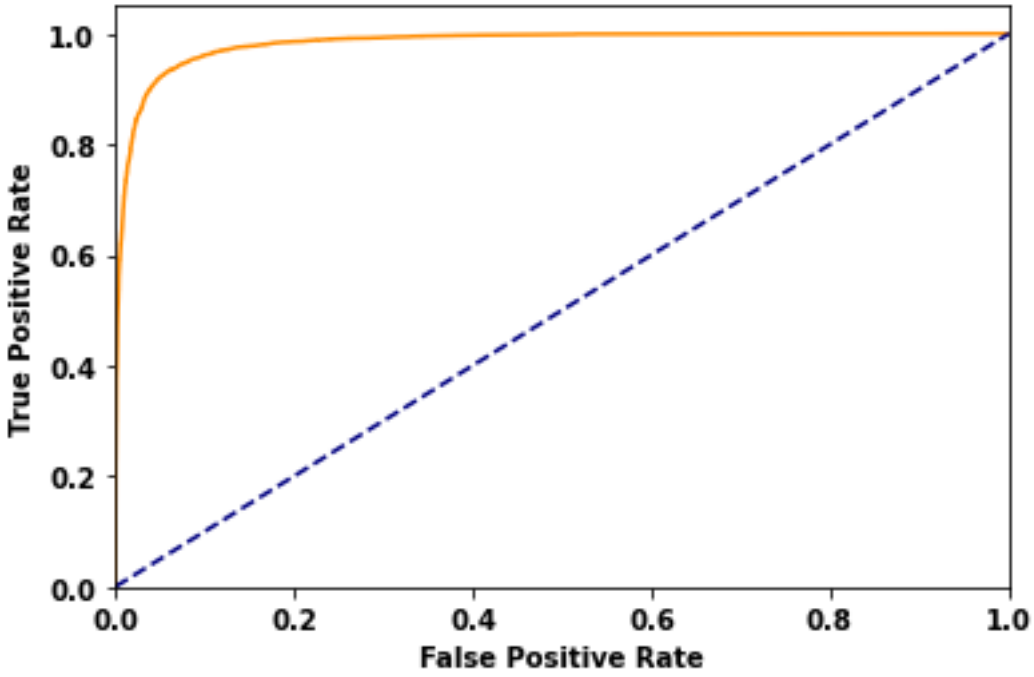
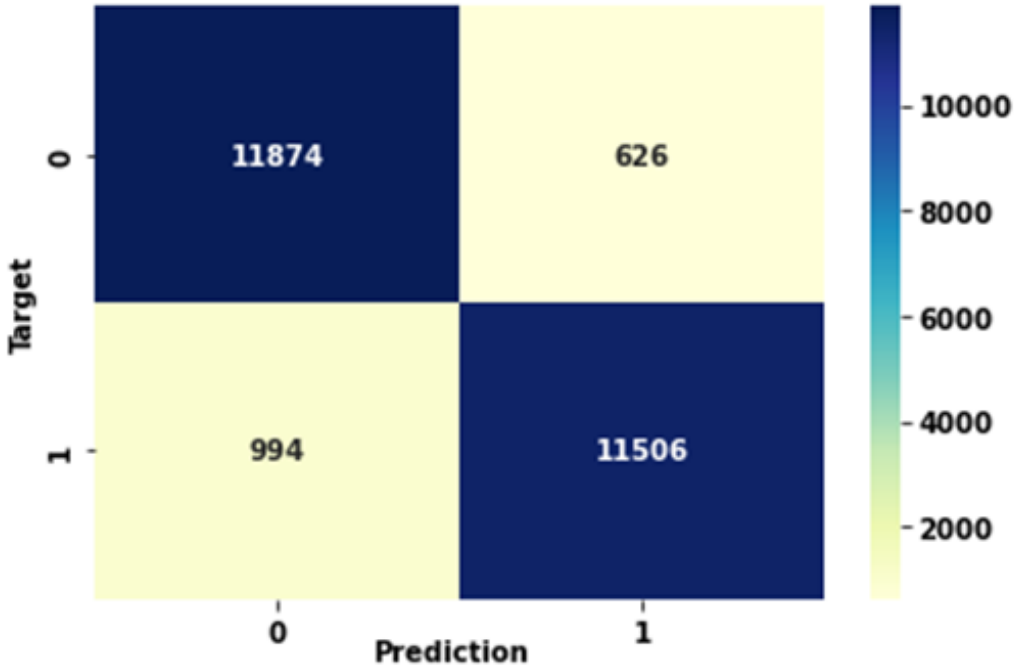


Figura 40 - Matriz de confusão para o conjunto de teste



Este modelo é tem o melhor resultado entre todos os modelos anteriores. Pode-se perceber também que ele tem uma facilidade maior com os comentários negativos, gerando assim menos falsos positivos do que falsos negativos.

Abaixo, os resultados dos comentários que o modelo tinha bastante certeza:

Figura 41 - Histograma com o resultados com previsão maior que 0,9

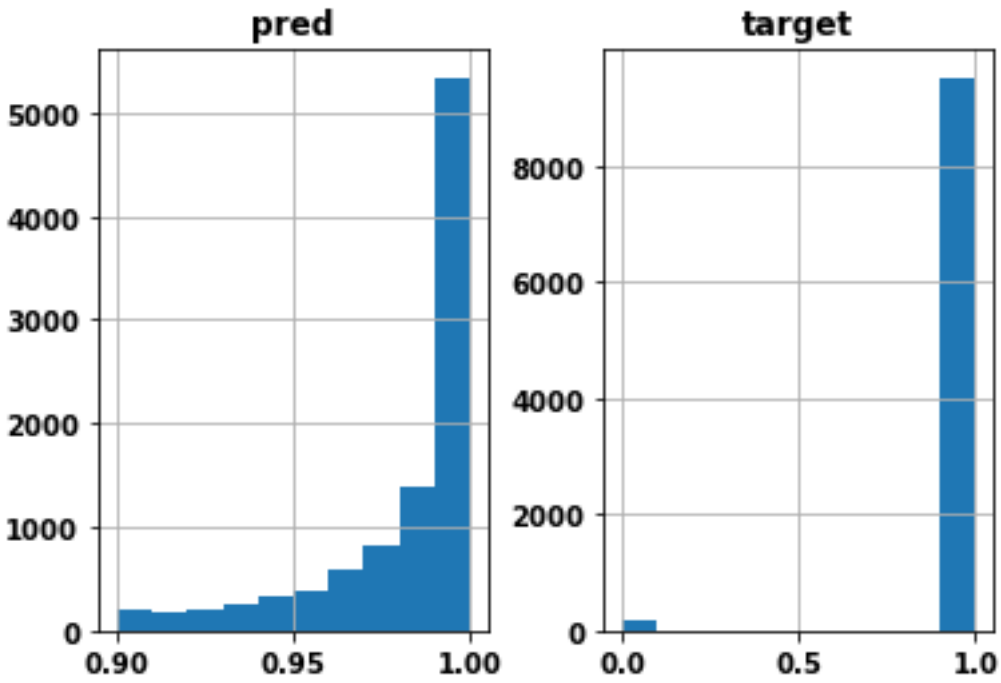
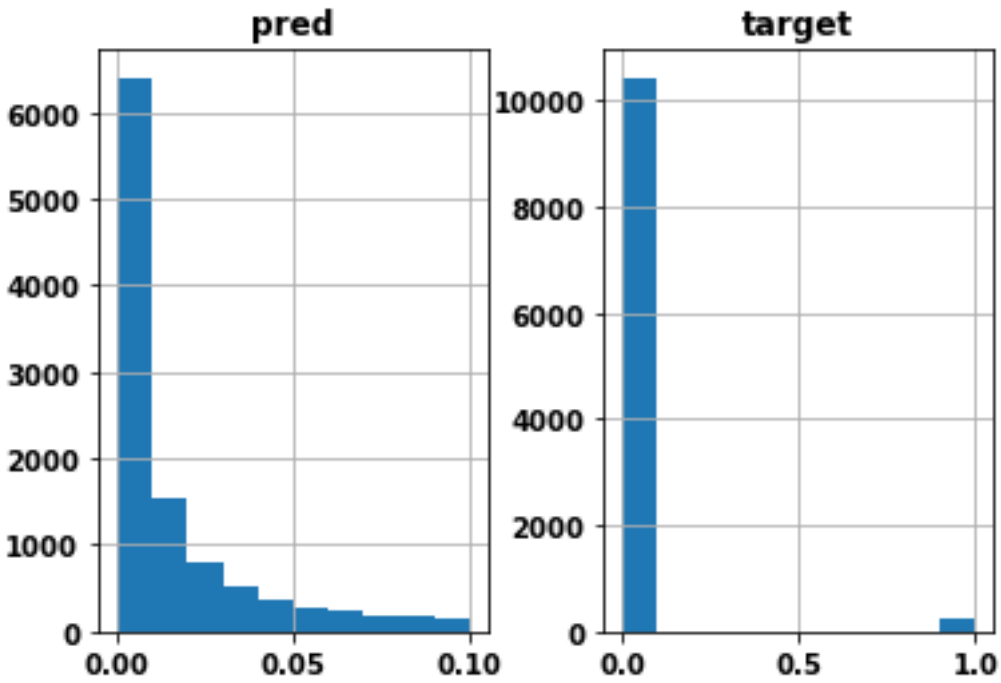


Figura 42 - Histograma com o resultados com previsão menor que 0,1



Verifica-se uma excelente performance também para esse modelo, gerando uma acurácia de 98,2% para os casos positivos e 97.7% para os casos negativos, mostrando mais uma vez que este modelo tem uma facilidade maior com os comentários negativos.

Na figura 43, são exibidos os comentários que o modelo tem os maiores erros no conjunto de treinamento:

Figura 43 - Maiores erros no treinamento

```
interp.show_top_losses(20)
```

Text	Prediction	Actual	Loss
xxbos xxmaj great movie - especially the music - xxmaj etta xxmaj james - " xxmaj at xxmaj last " . xxmaj this speaks volumes when you have finally found that special someone .	positive	negative	8.37
xxbos the photography was beautiful but i had difficulty understanding what was happening ... was there a lot of symbolism ? ... the 2 xxunk - do they mean something in xxmaj thai culture ? there 's not much plot , not much happens and it just meanders along . no real start , no real middle and no real end . rather unsatisfying really . xxmaj it was difficult	negative	positive	7.13
xxbos xxmaj the only reason i wanted to see this was because of xxmaj orlando xxmaj bloom . xxmaj simply put , the movie was spectacularly average . xxmaj it 's not bad , but it 's really not very good . xxmaj the editing is good ; the film is well - paced . xxmaj the direction is competent and assured . xxmaj the story is plodding . xxmaj	negative	positive	6.50
xxbos xxmaj after reading tons of good reviews about this movie i decided to take it for a spin (i bought it on xxup dvd , hence the " spin " pun ... xxmaj i'm a dork) . xxmaj the beginning was everything i hoped for , a perfect set - up (along with some quotes that i 've heard on xxmaj various xxmaj wu - xxmaj	negative	positive	5.54
xxbos i think it 's a great movie ! ! xxmaj it 's fun , maybe a little unrealistic , but fun and dramatic ! ! i would like to see it again , if they were showing it in xxup tv ! ! xxmaj just 1 question : xxmaj are we still talking about the same movie ? ? ?	positive	negative	5.30
xxbos normally i 'm not the sort to be scared by horror movies , but this movie is the exception . some how this movie got into my mind ! ! ! it is a very simple movie but at the same time xxunk effective , it has great atmosphere and this leads to some shocking moments , the girls father coming down the hill is a real standout .	positive	negative	5.27
xxbos xxmaj in the days before gore and sex took over , real horror films were made . xxmaj castle of xxmaj blood is , in my estimation , one of the finest , although other reviewers have given it mixed ratings . xxmaj in an odd sort of way it reminds of the more recent xxmaj the xxmaj others , which was in the theaters a couple of years	positive	negative	5.26
xxbos i read the novel ' xxmaj jane xxmaj eyre ' for the first time back in 1986 . xxmaj it was round that time that i saw the xxup bbc - version with xxmaj timothy xxmaj dalton and xxmaj zelah xxmaj clarke . xxmaj it was an excellent version and very much like the book . xxmaj years later , i laid eyes on this version and was horrified	positive	negative	5.05
xxbos xxmaj boston legal has turned its tail and is headed for the barn door and th pig slop it has created ! xxmaj when this show first aired almost four season back it was a humorous slap at the legal system which all actors seem to take pride in portraying . xxmaj it was funny , diversified , and to some extent factual . xxmaj the characters portrayed were	positive	negative	5.03
xxbos xxmaj this is the kind of film that , if it were made today , it would probably star xxmaj sandra xxmaj bullock and xxmaj hugh xxmaj grant ; actually , now that i think about it , this one is quite liable to be remade one day . xxmaj it 's pleasant , but with no depth whatsoever . xxmaj it suffers from the almost fatal miscasting of	negative	positive	4.99

Para esses modelos é possível visualizar as palavras que mais contribuem para a classificação do comentário. A seguir, apresentam-se quatro exemplos, um correto e positivo, um correto e negativo, um incorreto e positivo e outro incorreto e negativo. A escala de cores varia do menos para o mais importante: vermelho, laranja, amarelo e verde.

Figura 44 - Comentário positivo classificado corretamente pelo modelo

xxbos i went and saw this movie last night after being coaxed to by a few friends of mine . i 'll admit that i was reluctant to see it because from what i knew of xxmaj ashton xxmaj kutcher he was only able to do comedy . i was wrong . xxmaj kutcher played the character of xxmaj jake xxmaj fischer very well , and xxmaj kevin xxmaj costner played xxmaj ben xxmaj randall with such professionalism . xxmaj the sign of a good movie is that it can toy with our emotions . xxmaj this one did exactly that . xxmaj the entire theater (which was sold out) was overcome by laughter during the first half of the movie , and were moved to tears during the second half . xxmaj while exiting the theater i not only saw many women in tears , but many full grown men as well , trying desperately not to let anyone see them crying . xxmaj this movie was great , and i suggest that you go see it before you judge .

Figura 45 - Comentário negativo classificado corretamente pelo modelo

xxbos a funny thing happened to me while watching " xxmaj mosquito " : on the one hand , the hero is a deaf - mute and the director is totally unable to make us understand why he does what he does (mutilating mannequins ... er , excuse me , corpses) through his images . xxmaj on the other hand , the xxmaj english version at least is very badly dubbed . xxmaj so i found myself wishing there had been both more xxup and less dialogue at the same time ! xxmaj this film is stupid (funny how this guy has access to every graveyard and mortuary in his town) and lurid (where would we be in a 70s exploitationer without our gratuitous lesbian scene ?) . xxmaj not to mention the " romantic " aspect (oh , how xxunk ... xxmaj miss it . (*)

Figura 46 - Comentário negativo classificado incorretamente como positivo

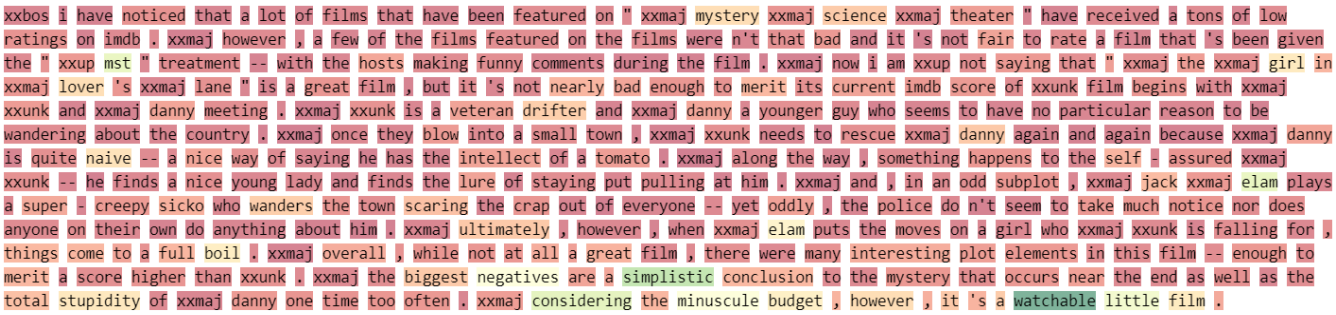
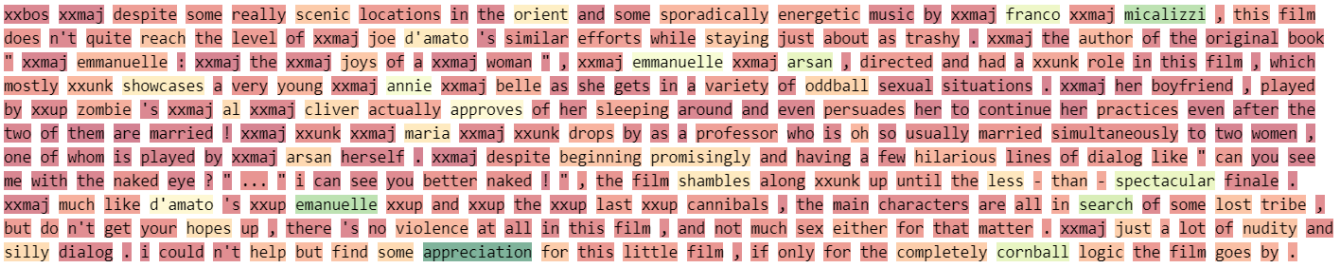


Figura 47 - Comentário positivo classificado incorretamente como negativo



Apesar de ser possível visualizar as palavras mais relevantes para a realização da classificação, a interpretação é bem mais superficial e de difícil interpretação do que no caso de bag of words.

7 Conclusão

Neste trabalho foi realizada uma análise de sentimentos de comentário no site do IMDb. Foi constatada a boa qualidade dos datasets encontrados, pois os mesmos tinham uma distribuição similar em todos as características analisadas.

Foram exercitados 3 métodos de aprendizado de máquina. O modelo de bag of words teve um bom resultado e ainda proporciona melhor explicabilidade do porquê da predição do modelo em dado comentário. O modelo de média das word embeddings pré-treinadas teve o pior resultado e ainda não proporciona nenhuma explicabilidade. Já a rede neural recorrente, por meio do *transfer learning,* teve a melhor performance entre todos os modelos e mostra as palavras mais importantes para a classificação.

8 Links

Segue os links para o repositório do github com todo o código gerado neste trabalho e do youtube com uma breve apresentação mostrando o trabalho:

Link para o vídeo: https://youtu.be/I6lsQFWF3hw
Link para o repositório: https://github.com/victorhbd/IMDb_sentiment