In [1]:
```
!pip3 install --upgrade scikit-learn
!pip3 install --upgrade python-decouple
!pip3 install --upgrade openai
!pip3 install --upgrade requests
```

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/site-packages (1.3.0)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.11/site-packages (from scikit
-learn) (1.25.0)
Requirement already satisfied: scipy>=1.5.0 in /usr/local/lib/python3.11/site-packages (from scikit-
learn) (1.11.0)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.11/site-packages (from scikit
-learn) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.11/site-packages (from
scikit-learn) (3.1.0)

[notice] A new release of pip is available: 23.0.1 -> 23.2.1
[notice] To update, run: /usr/local/opt/python@3.11/bin/python3.11 -m pip install --upgrade pip
Requirement already satisfied: python-decouple in /usr/local/lib/python3.11/site-packages (3.8)

[notice] A new release of pip is available: 23.0.1 -> 23.2.1
[notice] To update, run: /usr/local/opt/python@3.11/bin/python3.11 -m pip install --upgrade pip
Requirement already satisfied: openai in /usr/local/lib/python3.11/site-packages (0.27.8)
Requirement already satisfied: requests>=2.20 in /usr/local/lib/python3.11/site-packages (from opena
i) (2.31.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/site-packages (from openai) (4.65.
0)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.11/site-packages (from openai) (3.
8.5)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/site-packages
(from requests>=2.20->openai) (3.1.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/site-packages (from request
s>=2.20->openai) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/site-packages (from r
equests>=2.20->openai) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/site-packages (from r
equests>=2.20->openai) (2023.5.7)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.11/site-packages (from aiohtt
p->openai) (23.1.0)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.11/site-packages (from
aiohttp->openai) (6.0.4)
Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in /usr/local/lib/python3.11/site-package
s (from aiohttp->openai) (4.0.2)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.11/site-packages (from aioht
tp->openai) (1.9.2)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.11/site-packages (from ai
ohttp->openai) (1.4.0)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.11/site-packages (from aio
http->openai) (1.3.1)

[notice] A new release of pip is available: 23.0.1 -> 23.2.1
[notice] To update, run: /usr/local/opt/python@3.11/bin/python3.11 -m pip install --upgrade pip
Requirement already satisfied: requests in /usr/local/lib/python3.11/site-packages (2.31.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/site-packages
(from requests) (3.1.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/site-packages (from request
s) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/site-packages (from r
equests) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/site-packages (from r
equests) (2023.5.7)

[notice] A new release of pip is available: 23.0.1 -> 23.2.1
[notice] To update, run: /usr/local/opt/python@3.11/bin/python3.11 -m pip install --upgrade pip
```

In [11]:
```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import  LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

import pandas as pd
import os
import openai
import requests
import json
```

```
import numpy
from decouple import config
```

Pre Processing

In [20]:
```
def preprocessing(text,column_text):
    vectorizer = CountVectorizer(lowercase=False, max_features=50)
    bag_of_words = vectorizer.fit_transform(text[column_text])

    return bag_of_words

def classify_text_log(train_,test_,train_y_,test_y_):

    regression = LogisticRegression()
    regression.fit(train_,train_y_)
    return regression.score(test_,test_y_)

def classify_text_forest(train_,test_,train_y_,test_y_):
    rf = RandomForestClassifier()
    rf.fit(train_, train_y_)
    y_pred = rf.predict(test_)
    return accuracy_score(test_y_, y_pred)


def accurancy(actual, predict):
    return (actual == predict).mean()
```

In [13]:
```
review = pd.read_csv("./imdb/imdb-reviews-pt-br.csv")
classification = review["sentiment"].replace(["neg","pos"],[0,1])
review["classification"] = classification

review.head()
```

Out[13]:

	id	text_en	text_pt	sentiment	classification
0	1	Once again Mr. Costner has dragged out a movie...	Mais uma vez, o Sr. Costner arrumou um filme p...	neg	0
1	2	This is an example of why the majority of acti...	Este é um exemplo do motivo pelo qual a maiori...	neg	0
2	3	First of all I hate those moronic rappers, who...	Primeiro de tudo eu odeio esses raps imbecis, ...	neg	0
3	4	Not even the Beatles could write songs everyon...	Nem mesmo os Beatles puderam escrever músicas ...	neg	0
4	5	Brass pictures movies is not a fitting word fo...	Filmes de fotos de latão não é uma palavra apr...	neg	0

Modeling

Classifier Models

In [28]:
```
train, test, train_y, test_y = train_test_split(review, review.classification, random_state=3)

train_vetor = preprocessing(train,"text_pt")
test_vetor = preprocessing(test,"text_pt")

sample_log =  classify_text_log(train_vetor,test_vetor,train_y,test_y)
sample_forest = classify_text_forest(train_vetor,test_vetor,train_y,test_y)

#subsample with 1000 records
X_test_subsample1000, _, y_test_subsample1000, _ = train_test_split(test, test_y, train_size=1000, s
test_vetor_subsample1000 = preprocessing(X_test_subsample1000,"text_pt")

#subsample with 500 records
X_test_subsample500, _, y_test_subsample500, _ = train_test_split(test, test_y, train_size=500, stra
test_vetor_subsample500 = preprocessing(X_test_subsample500,"text_pt")

#subsample with 100 records
X_test_subsample, _, y_test_subsample, _ = train_test_split(test, test_y, train_size=100, stratify=t
test_vetor_subsample = preprocessing(X_test_subsample,"text_pt")
```

```
subsample_log1000 = classify_text_log(train_vetor,test_vetor_subsample1000,train_y,y_test_subsample
subsample_forest_1000 = classify_text_forest(train_vetor,test_vetor_subsample1000,train_y,y_test_su
subsample_log500 = classify_text_log(train_vetor,test_vetor_subsample500,train_y,y_test_subsample50
subsample_forest_500 = classify_text_forest(train_vetor,test_vetor_subsample500,train_y,y_test_subs
subsample_log100 = classify_text_log(train_vetor,test_vetor_subsample,train_y,y_test_subsample)
subsample_forest_100 = classify_text_forest(train_vetor,test_vetor_subsample,train_y,y_test_subsamp
```

Foram executados dois classificadores, nota-se que quando é utilizado uma amostra maior a acurácia é maior, contudo quando se vai reduzindo a amostra tem uma queda na acurácia, mas a pouca diferença entre o sample de 500 e 100. Devido a questão de custo optou-se pela subamostra de 100

Use Chat GPT With Classifier

In [34]:
```
API_KEY = config('OPENAI_API_KEY')
URL = "https://api.openai.com/v1/chat/completions"

message = """"""
vetor_result = []

headers = {
    "Content-Type":"application/json",
    "Authorization":f"Bearer {API_KEY}"
}



for i in range(0,100):
    message = """
Fazer uma análise de sentimento da avaliação do filme a seguir:

""" + X_test_subsample['text_pt'].iloc[i] + """

Classificar o essa avaliação como uma avaliação negativa, ou uma avaliação positiva, se for positiva
A saída desse ser somente o número, não deve constar nenhum texto, mais nada.
Exemplo de Saida:1, ou
Exemlo de Saida:0
"""
    data = {
    "model": "gpt-3.5-turbo",
    "messages": [{"role": "user", "content":message.strip()}],
    "temperature": 0.5,
    "max_tokens":20
    }
    response = requests.post(URL, headers=headers, data=json.dumps(data))
    response_data = response.json()
    message_content = int(response_data['choices'][0]['message']['content'])
    vetor_result.append(message_content)
    message = """"""
```

In [36]:
```
store_results = pd.DataFrame({'text_pt':X_test_subsample['text_pt'],
                              'classification_actual': X_test_subsample['classification'],
                              'classification_predict': vetor_result
                             })

sample_gpt = accuracy_score(store_results['classification_actual'],store_results['classification_pr
df_result_classify
```

In [38]:
```
label = ['Amostra Regressão Logistica',
         'Amostra Random Forest',
         'Subamostra Regressão Logistica (Sample 1000)',
         'Subamostra Random Forest (Sample 1000)',
         'Subamostra Regressão Logistica (Sample 500)',
         'Subamostra Random Forest (Sample 500)',
         'Subamostra Regressão Logistica (Sample 100)',
         'Subamostra Random Forest (Sample 100)',
         'Subamostra Chat GPT (Sample 100)'
        ]

values =  [sample_log,
           sample_forest,
           subsample_log1000,
           subsample_forest_1000,
           subsample_log500,
```

```
            subsample_forest_500,
            subsample_log100,
            subsample_forest_100,
            sample_gpt]

df_result_classify = pd.DataFrame({'label':label,'accuracy':values})
df_result_classify
```

Out[38]:

	label	accuracy
0	Amostra Regressão Logistica	0.651840
1	Amostra Random Forest	0.642863
2	Subamostra Regressão Logistica (Sample 1000)	0.662000
3	Subamostra Random Forest (Sample 1000)	0.657000
4	Subamostra Regressão Logistica (Sample 500)	0.546000
5	Subamostra Random Forest (Sample 500)	0.556000
6	Subamostra Regressão Logistica (Sample 100)	0.530000
7	Subamostra Random Forest (Sample 100)	0.540000
8	Subamostra Chat GPT (Sample 100)	0.960000