

Pra esse exercício comparei três algoritmos diferentes, usando como comparação métricas de avaliação de desempenho e matriz de confusão: Exercício um sem a utilização do GridSearchCV;

```
n_samples: 2000
      precision    recall  f1-score   support

     neg      0.86      0.80      0.83      240
     pos      0.82      0.88      0.85      260

 accuracy
macro avg      0.84      0.84      0.84      500
weighted avg      0.84      0.84      0.84      500

[[191  49]
 [ 31 229]]
```

Já no exercício dois com a utilização do GridSearchCV;

```
      precision    recall  f1-score   support

     neg      0.89      0.85      0.87      275
     pos      0.82      0.88      0.85      225

 accuracy
macro avg      0.86      0.86      0.86      500
weighted avg      0.86      0.86      0.86      500

[[233  42]
 [ 28 197]]
```

Agora no exercício três onde pede para utilizar algoritmo de classificação diferente foi usado o Naive Bayes e CountVectorizer;

```
n_samples: 2000
      precision    recall  f1-score   support

     neg      0.78      0.86      0.82      249
     pos      0.85      0.76      0.80      251

 accuracy
macro avg      0.81      0.81      0.81      500
weighted avg      0.81      0.81      0.81      500

[[214  35]
 [ 59 192]]
```

Os três modelos tiveram um desempenho muito próximo, mas é importante lembrar que a escolha do modelo mais adequado depende do contexto e dos objetivos específicos do problema. No caso da classificação de reviews de filmes em positivos e negativos, é fundamental minimizar os erros de classificação, evitando dar uma classificação negativa a um filme que é bom ou uma classificação positiva a um filme que é ruim.

Dito isso, considerando a acurácia como métrica de desempenho, o melhor modelo foi o que utilizou o GridSearchCV, que obteve uma acurácia de 86%. No entanto, é importante avaliar outras métricas, como a precisão e o recall, e também considerar outros fatores relevantes para o problema em questão, antes de decidir qual modelo utilizar.