```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
import pandas as pd
data = pd.read_csv("/content/drive/MyDrive/IMDB-Dataset.csv",index_col=None,encoding="latin-1")
data.head()
```

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. <br /><br />The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive

```
data.shape
```

```
(50000, 2)
```

```
X=data['review']
y=data['sentiment']
```

```
X.shape
```

```
(50000,)
```

```
y=y.replace({'positive':1,'negative':0})
#y=pd.get_dummies(data['sentiment'])# for multiclass problem
```

```
<ipython-input-6-8bf1a8d3a066>:1: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future version
  y=y.replace({'positive':1,'negative':0})
```

```
y
```

	sentiment
0	1
1	1
2	1
3	0
4	1
...	...
49995	1
49996	0
49997	0
49998	0
49999	0

50000 rows × 1 columns

dtype: int64

```
corpus=data['review']
```

```
lst_corpus = []
for string in corpus:
   lst_words = string.split()
   lst_grams = [" ".join(lst_words[i:i+1]) for i in range(0, len(lst_words), 1)]
```

```
    lst_corpus.append(lst_grams)
```

```
lst_corpus[0]
```

```
['One',
 'of',
 'the',
 'other',
 'reviewers',
 'has',
 'mentioned',
 'that',
 'after',
 'watching',
 'just',
 '1',
 'Oz',
 'episode',
 "you'll",
 'be',
 'hooked.',
 'They',
 'are',
 'right,',
 'as',
 'this',
 'is',
 'exactly',
 'what',
 'happened',
 'with',
 'me.<br',
 '/><br',
 '/>The',
 'first',
 'thing',
 'that',
 'struck',
 'me',
 'about',
 'Oz',
 'was',
 'its',
 'brutality',
 'and',
 'unflinching',
 'scenes',
 'of',
 'violence,',
 'which',
 'set',
 'in',
 'right',
 'from',
 'the',
 'word',
 'GO.',
 'Trust',
 'me,',
 'this',
 'is',
 'not',
```

```
len(lst_corpus)
```

```
50000
```

```
import gensim
word2vecobj = gensim.models.word2vec.Word2Vec(lst_corpus, vector_size=300,window=5, min_count=5, sg=0, epochs=30)
```

```
from tensorflow.keras import models, layers, preprocessing as kprocessing
tokenizer = kprocessing.text.Tokenizer(lower=True, split='',oov_token="NaN",)
tokenizer.fit_on_texts(lst_corpus)
dic_vocabulary = tokenizer.word_index
lst_sequences= tokenizer.texts_to_sequences(lst_corpus)
X_train_seq = kprocessing.sequence.pad_sequences(lst_sequences, maxlen=100, padding="post", truncating="post")
```

```
print(lst_sequences[0],end='\n')
```

```
[30, 5, 2, 80, 2223, 41, 1230, 11, 94, 149, 39, 845, 6088, 477, 418, 28, 11932, 33, 21, 2134, 15, 10, 7, 554, 48, 708, 16, 2424, 13, 93,
```

```
len(lst_sequences[0])
```

```
307
```

```
dic_vocabulary.items()
```

```
dict_items([('NaN', 1), ('the', 2), ('a', 3), ('and', 4), ('of', 5), ('to', 6), ('is', 7), ('in', 8), ('i', 9), ('this', 10),
('that', 11), ('it', 12), ('/><br', 13), ('was', 14), ('as', 15), ('with', 16), ('for', 17), ('but', 18), ('on', 19), ('movie', 20),
('are', 21), ('his', 22), ('not', 23), ('you', 24), ('film', 25), ('have', 26), ('he', 27), ('be', 28), ('at', 29), ('one', 30),
('by', 31), ('an', 32), ('they', 33), ('from', 34), ('all', 35), ('who', 36), ('like', 37), ('so', 38), ('just', 39), ('or', 40),
('has', 41), ('about', 42), ('her', 43), ("it's", 44), ('if', 45), ('some', 46), ('out', 47), ('what', 48), ('very', 49), ('when',
50), ('there', 51), ('more', 52), ('would', 53), ('even', 54), ('my', 55), ('good', 56), ('she', 57), ('their', 58), ('only', 59),
('no', 60), ('really', 61), ('had', 62), ('up', 63), ('can', 64), ('which', 65), ('see', 66), ('were', 67), ('than', 68), ('we', 69),
('-', 70), ('been', 71), ('get', 72), ('into', 73), ('will', 74), ('much', 75), ('because', 76), ('story', 77), ('how', 78), ('most',
79), ('other', 80), ('do', 81), ('also', 82), ("don't", 83), ('time', 84), ('its', 85), ('me', 86), ('great', 87), ('first', 88),
('make', 89), ('people', 90), ('could', 91), ('any', 92), ('/>the', 93), ('after', 94), ('then', 95), ('made', 96), ('bad', 97),
('think', 98), ('many', 99), ('being', 100), ('never', 101), ('him', 102), ('two', 103), ('<br', 104), ('too', 105), ('where', 106),
('little', 107), ('well', 108), ('watch', 109), ('way', 110), ('your', 111), ('it.', 112), ('did', 113), ('them', 114), ('know',
115), ('does', 116), ('movie.', 117), ('love', 118), ('best', 119), ('seen', 120), ('characters', 121), ('character', 122), ('these',
123), ('movies', 124), ('ever', 125), ('still', 126), ('over', 127), ('films', 128), ('plot', 129), ('such', 130), ('show', 131),
('acting', 132), ('should', 133), ('while', 134), ('those', 135), ('better', 136), ('off', 137), ('film.', 138), ('say', 139), ('go',
140), ('something', 141), ('why', 142), ('through', 143), ("doesn't", 144), ("didn't", 145), ("i'm", 146), ('scene', 147), ('makes',
148), ('watching', 149), ('film,', 150), ('movie,', 151), ('real', 152), ('find', 153), ('back', 154), ('actually', 155), ('scenes',
156), ('every', 157), ('few', 158), ('going', 159), ('man', 160), ('life', 161), ('same', 162), ('new', 163), ('/>i', 164),
('nothing', 165), ('look', 166), ('another', 167), ('lot', 168), ('quite', 169), ('thing', 170), ('&', 171), ('want', 172), ('end',
173), ('pretty', 174), ('old', 175), ('seems', 176), ("can't", 177), ('before', 178), ('got', 179), ('take', 180), ('actors', 181),
('give', 182), ('years', 183), ('part', 184), ('may', 185), ('young', 186), ('between', 187), ("that's", 188), ("i've", 189),
('both', 190), ('us', 191), ('without', 192), ('big', 193), ('thought', 194), ('things', 195), ('around', 196), ('it,', 197), ('now',
198), ('saw', 199), ('gets', 200), ('almost', 201), ('must', 202), ('though', 203), ('director', 204), ("isn't", 205), ('always',
206), ('here', 207), ('whole', 208), ('own', 209), ('come', 210), ('horror', 211), ('down', 212), ('work', 213), ('might', 214),
("there's", 215), ('"the', 216), ('cast', 217), ('am', 218), ("he's", 219), ('enough', 220), ('bit', 221), ('probably', 222),
('least', 223), ('feel', 224), ('last', 225), ('since', 226), ('long', 227), ('far', 228), ('funny', 229), ('kind', 230), ('each',
231), ('rather', 232), ('fact', 233), ('found', 234), ('original', 235), ('our', 236), ('world', 237), ('anything', 238), ('worst',
239), ('guy', 240), ('trying', 241), ('having', 242), ('interesting', 243), ('making', 244), ('done', 245), ('action', 246),
('comes', 247), ('right', 248), ('believe', 249), ('however,', 250), ('music', 251), ('anyone', 252), ('put', 253), ('main', 254),
('point', 255), ('played', 256), ('/>this', 257), ('goes', 258), ('worth', 259), ('hard', 260), ('looking', 261), ('role', 262),
('especially', 263), ('looks', 264), ('yet', 265), ("wasn't", 266), ('watched', 267), ('tv', 268), ('series', 269), ('during', 270),
('plays', 271), ('minutes', 272), ('family', 273), ('seem', 274), ('takes', 275), ('three', 276), ('someone', 277), ('performance',
278), ('script', 279), ('sure', 280), ('shows', 281), ('comedy', 282), ('different', 283), ('maybe', 284), ('everything', 285),
('although', 286), ('away', 287), ('set', 288), ('times', 289), ('time.', 290), ('woman', 291), ('left', 292), ('girl', 293),
('american', 294), ('seeing', 295), ('once', 296), ("you're", 297), ('simply', 298), ('fun', 299), ('completely', 300), ('play',
301), ('special', 302), ('everyone', 303), ('used', 304), ('john', 305), ('well,', 306), ('true', 307), ('again', 308), ('reason',
309), ('read', 310), ('high', 311), ('need', 312), ('until', 313), ('use', 314), ('black', 315), ('--', 316), ('idea', 317), ('dvd',
318), ('truly', 319), ('given', 320), ('sense', 321), ('beautiful', 322), ('nice', 323), ('recommend', 324), ('try', 325), ('place',
326), ('help', 327), ('came', 328), ('getting', 329), ('job', 330), ('rest', 331), ('version', 332), ('ending', 333), ('let', 334),
('excellent', 335), ('along', 336), ('keep', 337), ('half', 338), ('poor', 339), ('less', 340), ('full', 341), ('shot', 342),
('second', 343), ('couple', 344), ('tell', 345), ('money', 346), ('actor', 347), ('effects', 348), ('instead', 349), ('enjoy', 350),
('gives', 351), ('said', 352), ('(and', 353), ('audience', 354), ('definitely', 355), ('day', 356), ('understand', 357), ('fan',
358), ("couldn't", 359), ('playing', 360), ('went', 361), ('himself', 362), ('absolutely', 363), ('next', 364), ('early', 365),
('remember', 366), ('war', 367), ('book', 368), ('together', 369), ('entire', 370), ('certainly', 371), ('become', 372), ('small',
373), ('start', 374), ('screen', 375), ('supposed', 376), ('all,', 377), ('liked', 378), ('short', 379), ('several', 380), ('doing',
381), ('later', 382), ('felt', 383), ('2', 384), ('human', 385), ('loved', 386), ('often', 387), ('sort', 388), ('perhaps', 389),
('hollywood', 390), ('wife', 391), ('against', 392), ('men', 393), ('time,', 394), ('star', 395), ('(the', 396), ('totally', 397),
('night', 398), ('kids', 399), ('is,', 400), ('year', 401), ('seemed', 402), ('10', 403), ('piece', 404), ('production', 405),
('wonderful', 406), ('else', 407), ('.', 408), ('waste', 409), ('camera', 410), ('becomes', 411), ('top', 412), ('hope', 413),
('wanted', 414), ("she's", 415), ('able', 416), ('classic', 417), ("you'll", 418), ('home', 419), ('course', 420), ('based', 421),
('video', 422), ('called', 423), ('that,', 424), ('final', 425), ('death', 426), ('friends', 427), ('performances', 428), ('line',
429), ('women', 430), ('house', 431), ('them.', 432), ('â\x96', 433), ('live', 434), ('school', 435), ('mind', 436), ('lost', 437),
('name', 438), ('person', 439), ('wants', 440), ("i'd", 441), ('father', 442), ('perfect', 443), ('stupid', 444), ('gave', 445),
('sound', 446), ("they're", 447), ('under', 448), ('tries', 449), ('sex', 450), ('despite', 451), ('low', 452), ('turn', 453),
('story,', 454), ('one.', 455), ('already', 456), ('this,', 457), ("won't", 458), ('lead', 459), ('enjoyed', 460), ('either', 461),
('finally', 462), ('dead', 463), ('care', 464), ('budget', 465), ('turns', 466), ('guess', 467), ('this.', 468), ('mean', 469),
('written', 470), ('him.', 471), ('problem', 472), ('moments', 473), ('face', 474), ('lines', 475), ('took', 476), ('episode', 477),
('starts', 478), ('head', 479), ('favorite', 480), ('well.', 481), ('behind', 482), ('kill', 483), ('and,', 484), ('terrible', 485),
```

```
# Check if the word exists in the vocabulary before accessing it.
if 'ontip' in dic_vocabulary:
    index = dic_vocabulary['ontip']
    print(f"The index of 'ontip' is: {index}")
else:
    print(f"The word 'ontip' is not in the vocabulary.")
```

```
The word 'ontip' is not in the vocabulary.
```

```
import numpy as np
# embeddings is a matrix with zeros
embeddings = np.zeros((len(dic_vocabulary)+1, 300))
```

```
# dic_vocabulary.item() is a tuple with (word: indexvalue)
# in embedding location of index it is replaced with its corresponding embedding
for word,idx in dic_vocabulary.items():
    ## update the row with vector
    try:
        embeddings[idx] =  word2vecobj[word]
    ## if word not in model then skip and the row stays all 0s
    except:
        pass
```

```
type(X_train_seq)
```

```
numpy.ndarray
```

```
X_train_seq.shape
```

```
(50000, 100)
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X_train_seq, y, test_size=0.2, random_state=1000)
```

```
y_train_=pd.get_dummies(y_train)
y_test_=pd.get_dummies(y_test)
```

```
X_train.shape
```

```
(40000, 100)
```

```
vocab_size = len(tokenizer.word_index)+1
vocab_size
```

```
391794
```

```
from tensorflow.keras import layers
from tensorflow.keras.layers import  Input
```

```
embeddings.shape[0]# total number of words
```

```
391794
```

```
embeddings.shape[1]# vector size or embedding size
```

```
300
```

```
input = Input(shape=(100,))
x= layers.Embedding(input_dim=embeddings.shape[0],
                    output_dim=embeddings.shape[1],
                    weights=[embeddings],
                     trainable=False)(input)
```

```
x.shape
```

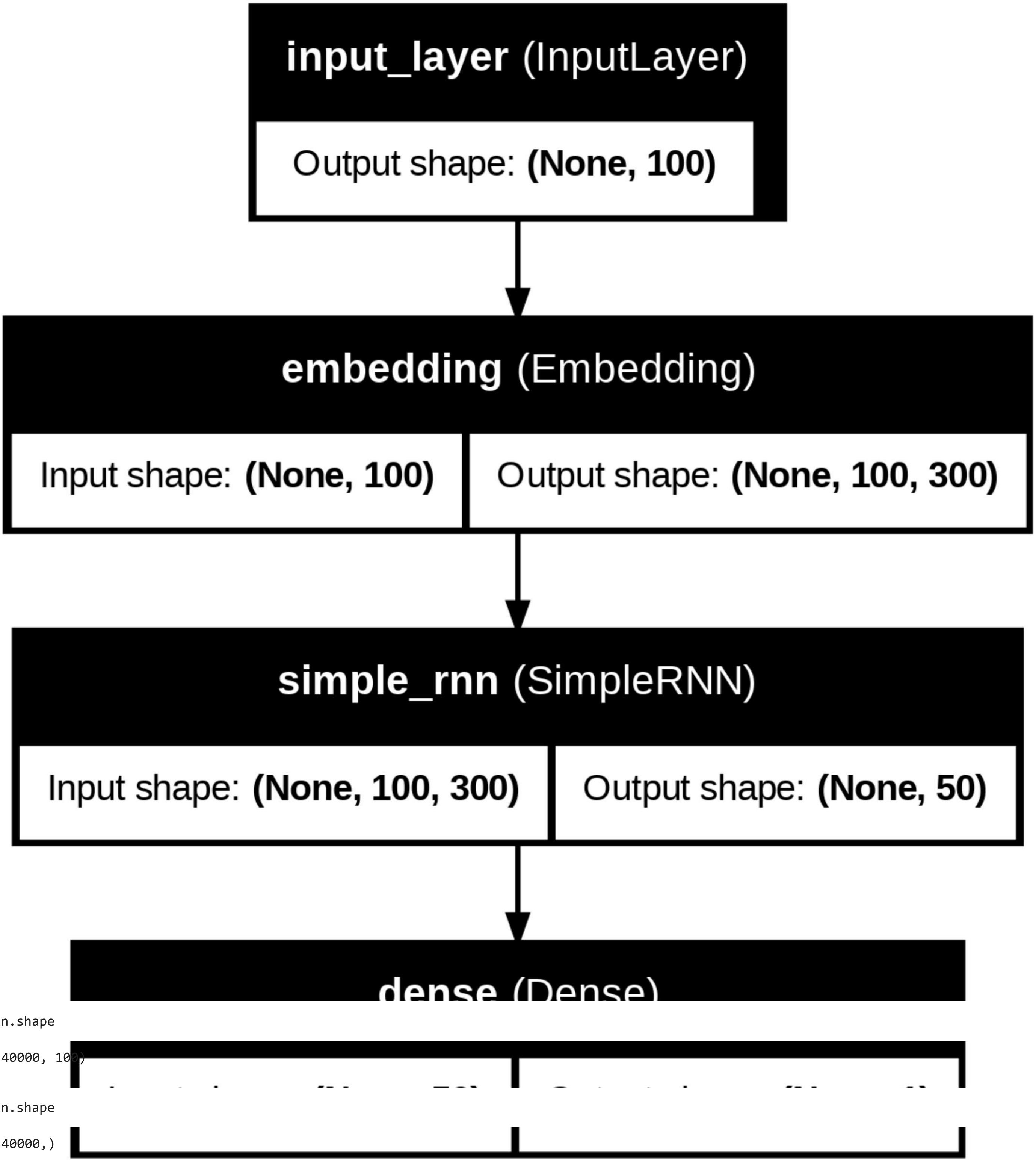
```
(None, 100, 300)
```

```
from tensorflow.keras.layers import Input,SimpleRNN,Dense
```

```
l1=SimpleRNN(50,return_sequences=False,activation='tanh')(x)
output=Dense(1,activation='sigmoid')(l1)
```

```
model=models.Model(input,output)
```

```
from tensorflow.keras.utils import plot_model
plot_model(model,show_shapes=True, show_layer_names=True)
```



```
X_train.shape
```

```
(40000, 100)
```

```
y_train.shape
```

```
(40000,)
```

```
model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
model.fit(X_train,y_train,batch_size=10,epochs=10,validation_split=0.1)
```

```
Epoch 1/10
3600/3600 ━━━━━━━━━━━━━━━━━━━━ 36s 10ms/step - accuracy: 0.4978 - loss: 0.6961 - val_accuracy: 0.4960 - val_loss: 0.6940
Epoch 2/10
3600/3600 ━━━━━━━━━━━━━━━━━━━━ 40s 9ms/step - accuracy: 0.5018 - loss: 0.6948 - val_accuracy: 0.4960 - val_loss: 0.6945
Epoch 3/10
3600/3600 ━━━━━━━━━━━━━━━━━━━━ 34s 10ms/step - accuracy: 0.4976 - loss: 0.6945 - val_accuracy: 0.5040 - val_loss: 0.6931
Epoch 4/10
3600/3600 ━━━━━━━━━━━━━━━━━━━━ 41s 10ms/step - accuracy: 0.5010 - loss: 0.6944 - val_accuracy: 0.5040 - val_loss: 0.6949
Epoch 5/10
3600/3600 ━━━━━━━━━━━━━━━━━━━━ 35s 10ms/step - accuracy: 0.4967 - loss: 0.6946 - val_accuracy: 0.5040 - val_loss: 0.6932
Epoch 6/10
3600/3600 ━━━━━━━━━━━━━━━━━━━━ 34s 9ms/step - accuracy: 0.4987 - loss: 0.6945 - val_accuracy: 0.4960 - val_loss: 0.6944
```

```
Epoch 7/10
3600/3600 ━━━━━━━━━━━━━━━━━━━━ 42s 10ms/step - accuracy: 0.5004 - loss: 0.6941 - val_accuracy: 0.4960 - val_loss: 0.6948
Epoch 8/10
3600/3600 ━━━━━━━━━━━━━━━━━━━━ 41s 10ms/step - accuracy: 0.4963 - loss: 0.6948 - val_accuracy: 0.4960 - val_loss: 0.6939
Epoch 9/10
3600/3600 ━━━━━━━━━━━━━━━━━━━━ 40s 9ms/step - accuracy: 0.5029 - loss: 0.6945 - val_accuracy: 0.5040 - val_loss: 0.6952
Epoch 10/10
3600/3600 ━━━━━━━━━━━━━━━━━━━━ 42s 10ms/step - accuracy: 0.5010 - loss: 0.6944 - val_accuracy: 0.4960 - val_loss: 0.6940
<keras.src.callbacks.history.History at 0x781a05a1bf50>
```

```
res=model.predict(X_test)
```

```
313/313 ━━━━━━━━━━━━━━━━━━━━ 2s 5ms/step
```

```
rounded = [round(x[0]) for x in res]
```

```
from sklearn.metrics import classification_report, confusion_matrix
# Print the classification report
print(classification_report(y_test, rounded))
```

```
              precision    recall  f1-score   support

           0       0.51      1.00      0.67      5054
           1       0.00      0.00      0.00      4946

    accuracy                           0.51     10000
   macro avg       0.25      0.50      0.34     10000
weighted avg       0.26      0.51      0.34     10000

/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and be
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and be
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and be
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```