HATE SPEECH IDENTIFICATION-TRISHAA

PIPELINE FOR SENTIMENT ANLYSIS TASK

1. Data Collection: Gather labeled data (e.g., reviews, tweets) for sentiment analysis.

2. Data Preprocessing: Clean and preprocess data by removing noise, tokenizing, lowercasing, removing stopwords, and performing stemming/lemmatization.

3. Feature Extraction: Convert text data into numerical features using techniques like Bag-of-Words, TF-IDF, or word embeddings.

4. Model Selection: Choose a suitable model such as Logistic Regression, Naive Bayes, SVM, RNN, CNN, or Transformer-based models like BERT.

5. Model Training: Split data into training and testing sets, train the model on the training data, and tune hyperparameters if necessary.

6. Model Evaluation: Evaluate the trained model's performance on the testing data using metrics like accuracy, precision, recall, and F1-score.

7. Fine-tuning: Refine the model by adjusting hyperparameters or experimenting with different architectures.

8. Deployment: Deploy the trained model into production, integrating it with other systems or applications.

```
# Import necessary libraries
import numpy as np
import pandas as pd
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from gensim.models import Word2Vec, FastText
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Embedding, Conv1D, MaxPooling1D, LSTM, Dense


# Download NLTK resources
nltk.download('stopwords')
nltk.download('punkt')

# Load hate speech dataset (assuming it's in CSV format)
data = pd.read_csv('/labeled_data.csv')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
```

```
# Data Cleaning and Preprocessing
def preprocess_text(text):
    text = text.lower()  # Convert to lowercase
    text = word_tokenize(text)  # Tokenization
    text = [word for word in text if word.isalnum()]  # Remove non-
alphanumeric characters
    text = [word for word in text if word not in
stopwords.words('english')]  # Remove stopwords
    return text

data['clean_text'] = data['tweet'].apply(preprocess_text)


data['clean_text']
```

```
0        [rt, mayasolovely, woman, complain, cleaning, ...
1        [rt, mleew17, boy, dats, cold, tyga, dwn, bad,...
2        [rt, urkindofbrand, dawg, rt, 80sbaby4life, ev...
3                                [rt, look, like, tranny]
4        [rt, shenikaroberts, shit, hear, might, true, ...
                               ...
24778    [muthaf, lie, 8220, lifeasking, right, tl, tra...
24779    [gone, broke, wrong, heart, baby, drove, redne...
24780    [young, buck, wan, na, eat, dat, nigguh, like,...
24781              [youu, got, wild, bitches, tellin, lies]
24782    [ntac, eileen, dahlia, beautiful, color, combi...
Name: clean_text, Length: 24783, dtype: object
```

```
# Split data into training and testing sets
X_train, X_test, y_train, y_test =
train_test_split(data['clean_text'], data['class'], test_size=0.2,
random_state=42)
```

WORD2VEC

```
# Word2Vec
word2vec_model = Word2Vec(sentences=X_train, vector_size=100,
window=5, min_count=1, workers=4)
X_train_word2vec = np.array([np.mean([word2vec_model.wv[word] for word
in words if word in word2vec_model.wv] or [np.zeros(100)], axis=0) for
words in X_train])
X_test_word2vec = np.array([np.mean([word2vec_model.wv[word] for word
in words if word in word2vec_model.wv] or [np.zeros(100)], axis=0) for
words in X_test])
```

```
print(word2vec_model)
```

```
Word2Vec<vocab=25473, vector_size=100, alpha=0.025>
```

```
from sklearn.metrics import confusion_matrix, classification_report

# Predict labels for Word2Vec + Logistic Regression
y_pred_word2vec = logreg_word2vec.predict(X_test_word2vec)

# Evaluate Word2Vec + Logistic Regression Accuracy
word2vec_accuracy = accuracy_score(y_test, y_pred_word2vec)
print("Word2Vec + Logistic Regression Accuracy:", word2vec_accuracy)
```

```
Word2Vec + Logistic Regression Accuracy: 0.8355860399435142
```

```
  # Print Word Representations from Word2Vec
print("Word Representations from Word2Vec:")
for word, representation in zip(word2vec_model.wv.index_to_key[:10],
word2vec_model.wv.vectors[:10]):
    print(word, representation)
```

```
Word Representations from Word2Vec:
bitch [-0.30385545  1.3490597   0.24555592  0.37775442 -0.24384387 -
2.0314631
  1.1677946   2.4828095  -0.67582285 -0.7674058  -0.18649688 -
1.5680487
 -0.3430366   0.53939945 -0.07178079 -1.1282406   0.6195518  -
0.6651011
 -0.05511921 -2.1791525   0.5110197   0.7341875   1.1283576  -
0.23003629
 -0.23346585  0.34712282 -0.62616545 -0.57454044 -1.2170774  -
0.1798486
  0.990339    0.24647652  0.35013536 -1.3273172  -0.6314316
1.0967523
  0.48773274 -0.62207764 -1.0021445  -2.1685905   0.3221075  -
0.5910369
 -0.5623844  -0.20302433  0.1809369  -0.7670203  -1.0036834  -
0.10144753
  0.1408536   1.028045    0.42124674 -0.6486839  -0.1117289  -
0.41965684
 -0.4530616   0.701973    1.0463613  -0.406821   -1.1287614
0.5009816
  0.07095814  0.3114     -0.04395135 -0.10771189 -1.0333594
1.2207897
  0.46354604  0.8860421  -2.0075004   1.3416872  -0.45429212
0.5074623
  1.1953722  -0.9097047   1.1476848   0.17832312  0.6698421
0.0241885
 -0.7836555   0.39675346 -0.79276884 -0.0975917  -1.0195584
```

```
1.2289559
 -0.43986467 -0.23350126  0.3945025   1.1166371   1.3307644
0.4178626
  1.3252807   0.9183199   0.34948617  0.08292788  2.2053773
0.9294945
  0.3547289  -1.1556379  -0.15733257  0.141734  ]
rt [-0.10902284  1.44108     0.0399925   0.47080168 -0.3737661  -
2.1144013
  1.2345134   2.8280861  -0.67078626 -0.7439037  -0.40934268 -
1.5709552
 -0.24003066  0.5715537  -0.19214733 -1.2131616   0.91290116 -
0.6268258
  0.17874491 -2.766372    0.5700779   0.7045212   1.4395218  -
0.23515582
 -0.14267041 -0.02232146 -0.6915171  -0.8477867  -1.3930222  -
0.04851014
  1.4476947   0.25035703  0.5518011  -1.6927906  -0.8280136
1.5883955
  0.43295667 -0.713608   -1.2395118  -2.6100934   0.4420774  -
0.7738093
 -0.7417349  -0.181047    0.26385236 -0.7740492  -1.1867809  -
0.3147959
  0.01706811  1.2595229   0.37206474 -0.78103715 -0.228213   -
0.45146206
 -0.6793369   0.77076596  1.0063348  -0.5185122  -1.3081862
0.5897733
 -0.00417105  0.27469957 -0.15929389 -0.16794224 -1.3407167
1.2743475
  0.76769674  0.7343948  -2.4285686   1.8686739  -0.5354626
0.44980124
  1.316182   -1.0629877   1.3711854  -0.02209738  0.6840473  -
0.04180627
 -0.9774079   0.58735085 -0.91965306 -0.33882838 -1.4889277
1.4777077
 -0.65360034 -0.31088108  0.44371206  1.2958547   1.5134841
0.3522306
  1.4988229   0.9445347   0.23754697  0.18227993  2.2774765
0.99294466
  0.19798054 -1.6579951  -0.04216954  0.2305862 ]
128514 [ 0.3712174   1.4668155  -0.9713205   0.8403932   0.16255419 -
1.1180954
  0.97293246  2.2428796  -1.4118183   0.00377866 -0.41414145 -
2.1668828
  1.1518475   0.9803033  -0.00371499 -0.77308965  1.9264969  -
0.03889432
 -0.26972252 -2.6744227   0.2752213  -0.04942014  1.1578182  -
0.19448441
  0.95619035 -0.5230038  -0.24589795 -0.2882834  -1.7554973
0.6157875
```

```
  2.1966422   0.20343806  0.17273659 -2.2544358  -1.9404454
1.1295233
  0.72123545 -0.65740144 -1.682882   -2.547804   -1.0692164
0.22149326
 -0.48652133  0.02362313  0.4948911  -0.107682   -1.3469493  -
0.9356886
 -0.4418177   1.7485569   0.12056325 -0.8883503  -1.2111617  -
1.5552828
 -1.1848003  -0.25925824  0.6896296  -0.7571202  -1.9278108
0.7446554
  0.33601555 -0.9222964   0.32134807 -0.12335879 -2.3292005   1.931389
  0.8603723   0.7397222  -1.7910782   2.4445314   0.10850294
0.42418602
  1.8183154  -0.32566047  1.014446    0.21230812  1.4226425
0.2727441
 -1.666877    0.23659381 -0.7721388  -0.9893223  -2.2427878
1.6471484
 -1.195812   -1.1247046   0.23219644  0.5324418   1.7739854
0.04836475
  1.0102934   1.7415271   0.5314235  -0.60100347  2.587723
1.2345011
  0.44978586 -2.0516858   0.994727    0.11819554]
bitches [-0.18554868  1.336247    0.08732136  0.44121373 -0.26604068 -
1.9168979
  1.1710919   2.4660175  -0.5970824  -0.7399921  -0.21183799 -
1.4158084
 -0.25176388  0.49678138 -0.10659748 -1.0803332   0.7847168  -
0.6513702
  0.01368743 -2.1779842   0.5002187   0.6879478   1.1684488  -
0.17112713
 -0.24828118  0.11911512 -0.6243034  -0.64268315 -1.1649226  -
0.11987352
  1.1238358   0.28650743  0.39397573 -1.3633393  -0.6708089
1.2800397
  0.44244757 -0.69307584 -1.1024063  -2.1966393   0.37268227 -
0.6023683
 -0.5245132  -0.25117412  0.25084144 -0.66920507 -1.0040276  -
0.2204487
  0.10262871  1.0277792   0.3965063  -0.65107495 -0.23774897 -
0.43053553
 -0.57672143  0.6659825   0.98887205 -0.35441092 -1.0897284
0.39287478
  0.05788196  0.3134859  -0.03581379 -0.13725074 -1.07669
1.1284243
  0.4603338   0.7851106  -2.0569355   1.4250332  -0.38198543
0.43274036
  1.1968957  -0.8709176   1.1540294   0.005722    0.6120267
0.0695221
 -0.84051704  0.45033982 -0.81332296 -0.17781192 -1.1204766
```

```
1.2708025
 -0.45895332 -0.2986056   0.42293426  1.153337    1.3024999
0.29130125
  1.3023926   0.8539193   0.2615455   0.11264548  2.0803428   0.845906
  0.31266913 -1.3223761  -0.09824669  0.18634567]
http [ 1.7240107e-01  1.1077211e+00 -3.6559507e-01  5.8046550e-01
 -3.0856109e-01 -1.3772324e+00  1.0115252e+00  2.3116965e+00
 -7.6398313e-01 -3.7337095e-01 -5.7383335e-01 -1.4236895e+00
  2.9426298e-01  5.8335686e-01 -2.6536757e-01 -9.4271922e-01
  1.1570984e+00 -2.8755766e-01  1.6925193e-01 -2.5822132e+00
  4.5144007e-01  2.9898548e-01  1.3002012e+00 -1.2762541e-01
  2.8469849e-01 -4.3290204e-01 -4.6086243e-01 -6.6932315e-01
 -1.3216592e+00  3.3709288e-01  1.5775390e+00  1.5478253e-01
  4.7575027e-01 -1.7141838e+00 -1.1375906e+00  1.4451454e+00
  4.0797693e-01 -6.6270530e-01 -1.3606083e+00 -2.3446984e+00
 -2.7240343e-02 -4.3029377e-01 -7.1496272e-01  2.4888418e-03
  3.7792438e-01 -5.7825226e-01 -1.0779127e+00 -6.4299953e-01
 -1.2576470e-01  1.3406775e+00  1.4956561e-01 -6.2555552e-01
 -4.3760228e-01 -6.9809997e-01 -7.6581997e-01  2.8415838e-01
  5.5995429e-01 -7.0780665e-01 -1.3209211e+00  5.2257133e-01
 -1.3841846e-02 -1.6060543e-01 -1.3604204e-01 -2.2819790e-01
 -1.4430670e+00  1.2324985e+00  9.2544621e-01  5.7656902e-01
 -2.0691099e+00  1.9561603e+00 -4.0000248e-01  4.4379532e-01
  1.1574787e+00 -8.1197435e-01  1.1908638e+00 -1.7487893e-02
  9.3607002e-01  6.4658793e-03 -1.1637543e+00  4.3505347e-01
 -8.1227332e-01 -7.4798185e-01 -1.7103851e+00  1.3945760e+00
 -9.6902680e-01 -5.7294512e-01  3.2430574e-01  8.3230281e-01
  1.2239207e+00  2.2969541e-01  1.1750958e+00  9.4341820e-01
  1.5993893e-01  7.5773649e-02  2.0397129e+00  9.3357557e-01
  9.6657537e-02 -1.8008951e+00  4.3459806e-01  1.8760559e-01]
like [-0.2809391   1.3416367   0.16565472  0.3050463  -0.35466868 -
1.9792504
  1.1237191   2.4845688  -0.6306865  -0.78002405 -0.27085578 -1.475388
 -0.3341602   0.5026878  -0.13503328 -1.162278    0.61155885 -
0.65057373
  0.07517727 -2.2088099   0.47360298  0.66071135  1.2046514  -
0.22345747
 -0.29298908  0.20731838 -0.6275488  -0.7257385  -1.1719967  -
0.09278905
  1.0873432   0.19067575  0.3386845  -1.4536431  -0.5782264
1.1972328
  0.38943744 -0.57275456 -0.93754137 -2.1935093   0.47999763 -
0.66150945
 -0.60022545 -0.24830897  0.19008505 -0.73634315 -1.012887    -
0.12917145
  0.07570811  0.954072    0.42668712 -0.72571445 -0.04187254 -
0.3358179
 -0.53065723  0.7917871   0.9982505  -0.3991702  -1.1427553
0.44320557
```

```
  0.0981284   0.291323   -0.08654553 -0.0907158  -1.0824217
1.1596721
  0.51844597  0.7980497  -2.0791516   1.4139591  -0.4368066
0.49203232
  1.2094631  -0.89205104  1.1774979   0.1539235   0.48507893
0.05630538
 -0.75043947  0.54420984 -0.76467615 -0.1960049  -1.0764841
1.2085217
 -0.40109986 -0.23286912  0.42408645  1.1884995   1.3641737
0.37225017
  1.4231731   0.86591375  0.17383336  0.1104614   2.04085
0.87557346
  0.26904    -1.2909025  -0.16707444  0.1692897 ]
hoes [-0.11233591  1.2873642   0.01520295  0.44829994 -0.16401961 -
1.8082011
  1.0736461   2.3077242  -0.6355134  -0.61568254 -0.2320802  -
1.4301171
 -0.20931849  0.5340611  -0.10599998 -0.9956422   0.7578387  -
0.61968017
 -0.03682573 -2.079294    0.48884004  0.6811439   1.0614536  -
0.1392081
 -0.14836627  0.12751274 -0.6113835  -0.522354   -1.1229969  -
0.05429415
  1.1108044   0.3176913   0.33736977 -1.3565876  -0.76713806
1.1407795
  0.5146893  -0.6396694  -1.0966085  -2.0630908   0.23696119 -
0.49019888
 -0.43725672 -0.23599587  0.27392992 -0.6037575  -0.9801234  -
0.29698864
  0.14129278  1.0038595   0.3529027  -0.6462708  -0.31557903 -
0.38454247
 -0.51933295  0.5601953   0.99239177 -0.38703606 -1.1162919
0.33988997
  0.01573452  0.18599737 -0.00678394 -0.1336507  -1.1430901
1.1650356
  0.39239606  0.6859893  -1.8932465   1.406987   -0.3641093
0.4146624
  1.1827222  -0.7299565   1.0726949   0.07849014  0.697467
0.11015215
 -0.8530441   0.32350236 -0.7451665  -0.20288546 -1.1110742
1.2268897
 -0.4688723  -0.3101972   0.3508536   1.0960182   1.2571636
0.3386958
  1.2019235   0.8990935   0.24477534  0.00732843  2.0212069
0.8022401
  0.36586517 -1.2057091  -0.0459452   0.18170203]
pussy [-0.09888324  1.1972934   0.17019174  0.41151217 -0.3663847  -
1.8687587
  1.113094    2.4500313  -0.57943815 -0.70374876 -0.3448926  -
```

```
1.3354315
 -0.28738385  0.4302314  -0.16899286 -1.0751046   0.68269956 -0.588408
  0.07735557 -2.291314    0.4971891   0.70327705  1.2182686  -
0.21991369
 -0.17489268  0.12310491 -0.6183506  -0.7263009  -1.1873584  -
0.08210324
  1.0909622   0.26238683  0.44969425 -1.3932356  -0.6419948
1.3373486
  0.32734525 -0.6773675  -1.0903143  -2.2518766   0.5037658  -
0.72643465
 -0.64690864 -0.11176544  0.23228481 -0.7750669  -1.0128078  -
0.22351567
  0.08108915  1.0448236   0.3153326  -0.631653   -0.09866951 -
0.34626174
 -0.5655189   0.7664236   0.81781787 -0.44818437 -1.0856425
0.47160015
  0.04438299  0.31929988 -0.16112238 -0.09717815 -1.0451857
1.1339351
  0.6089718   0.7331258  -2.1180038   1.4687719  -0.5085466
0.45799246
  1.0593309  -0.97663873  1.1727794   0.03840374  0.62338287 -
0.05721272
 -0.7958571   0.50964993 -0.79188955 -0.24908966 -1.1963713   1.213891
 -0.53670794 -0.17547132  0.42190725  1.1811688   1.2555104
0.33983308
  1.3294393   0.74316275  0.20492174  0.27231854  1.9746933   0.845565
  0.18355288 -1.3557972  -0.07167249  0.15865164]
hoe [-0.16353402  1.2858324   0.07992818  0.41322362 -0.19055006 -
1.7963297
  1.0592047   2.3125687  -0.6408043  -0.5872092  -0.26748887 -1.48652
 -0.19659595  0.5080132  -0.10542452 -1.0558747   0.70172465 -
0.5895044
  0.01543099 -2.1415129   0.4354328   0.6389185   1.1326295  -
0.18885078
 -0.10450581  0.14850591 -0.59289896 -0.61444974 -1.2091855  -
0.09291403
  1.1406916   0.18262045  0.33478284 -1.3424735  -0.74863136
1.1068609
  0.437774   -0.5577818  -1.0045879  -2.1545837   0.21471524 -
0.55390966
 -0.590336   -0.1468698   0.19921069 -0.695208   -0.98268706 -
0.23455581
  0.08057723  1.0372993   0.3214159  -0.65979844 -0.16306837 -
0.4561178
 -0.49742877  0.59347105  0.9509863  -0.45040923 -1.118192
0.4793658
 -0.00760022  0.14548866 -0.04402552 -0.10639887 -1.1118613
1.1453884
  0.55143046  0.7228674  -1.9698471   1.4717792  -0.4637928
```

```
0.44885042
  1.1484909  -0.84251195  1.0868828   0.10725488  0.6368786  -
0.01433526
 -0.8339927   0.3616677  -0.7744186  -0.24735196 -1.1381232
1.2000579
 -0.5152289  -0.3043924   0.32536134  1.0415325   1.2895768
0.3601159
  1.2285168   0.88711315  0.28002554  0.04738237  2.0593226
0.9195342
  0.2970658  -1.2168735  -0.05109182  0.12005351]
8220 [ 0.09597573  1.2340666  -0.34186253  0.5826111  -0.23158379 -
1.5357348
  1.1070557   2.3659909  -0.80210096 -0.45033684 -0.40334082 -
1.4960741
  0.28583544  0.66762906 -0.17107812 -0.8140855   1.2425332  -
0.43868908
  0.03348907 -2.5839946   0.58748555  0.3635334   1.2019539  -
0.17571914
  0.22073469 -0.27474952 -0.53516066 -0.54628456 -1.2677162
0.22270556
  1.5958518   0.32343537  0.5699789  -1.7226418  -1.0487986
1.4380959
  0.5121059  -0.6637206  -1.3928586  -2.2675345  -0.02704727 -
0.37680858
 -0.5786707  -0.12275697  0.35632452 -0.43709552 -1.1984686  -
0.47044832
 -0.12730832  1.3909115   0.25080884 -0.691864   -0.65351367 -
0.65502805
 -0.72580504  0.3519584   0.6828737  -0.48832342 -1.3091457
0.53700566
  0.10953099 -0.06113624  0.01446497 -0.14557526 -1.5375254
1.2809175
  0.6507035   0.62157345 -1.9006778   1.8690566  -0.23610173
0.40033647
  1.3011737  -0.69263285  1.106282   -0.09840566  0.90801466
0.03645724
 -1.1621547   0.4883882  -0.7391603  -0.56848323 -1.5477446
1.5147961
 -0.8463669  -0.5640761   0.271923    0.9257704   1.3448598
0.2044141
  1.1962929   1.1058495   0.28146854 -0.01162046  2.112072
0.75867844
  0.1421498  -1.6999454   0.39570814  0.29131973]

# Analyze misclassified instances for Word2Vec + Logistic Regression
misclassified_word2vec = X_test[y_test != y_pred_word2vec]
true_labels_word2vec = y_test[y_test != y_pred_word2vec]
predicted_labels_word2vec = y_pred_word2vec[y_test != y_pred_word2vec]
misclassified_df_word2vec = pd.DataFrame({'Text':
misclassified_word2vec, 'True Label': true_labels_word2vec, 'Predicted
```

```
Label': predicted_labels_word2vec})
print("Misclassified instances for Word2Vec + Logistic Regression:")
print(misclassified_df_word2vec)


Misclassified instances for Word2Vec + Logistic Regression:
                                                    Text  True
Label  \
18943  [rt, 1inkkofrosess, lol, credit, ai, near, goo...           2

4273            [search, gay, redneck, episode, 1, play]           0

3778   [keebithalal, loganswarning, got, ta, love, is...           0

15789  [rt, jsu, coach, omar, johnson, u, ball, u, th...           2

11311  [tryna, get, sleep, birds, start, getting, rowdy]           2

...                                                  ...         ...

4767   [stevestockmantx, hes, friggin, idiot, say, an...           0

10959                        [think, eat, brownie, pass]           2

20979  [real, unreal, lol, yankees, worldseries, 27an...           2

7339                       [xcorey21, uh, trash, 128536]           1

3310   [grizzboadams, wyattnuckels, haha, ight, nig, ...           0


       Predicted Label
18943                1
4273                 2
3778                 1
15789                1
11311                1
...                ...
4767                 1
10959                1
20979                1
7339                 2
3310                 2

[815 rows x 3 columns]

# Generate confusion matrix and classification report for Word2Vec +
Logistic Regression
print("Confusion Matrix for Word2Vec + Logistic Regression:")
print(confusion_matrix(y_test, y_pred_word2vec))
```

```python
print("\nClassification Report for Word2Vec + Logistic Regression:")
print(classification_report(y_test, y_pred_word2vec))
```

```
Confusion Matrix for Word2Vec + Logistic Regression:
[[   0  220   70]
 [   0 3674  158]
 [   0  367  468]]

Classification Report for Word2Vec + Logistic Regression:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       290
           1       0.86      0.96      0.91      3832
           2       0.67      0.56      0.61       835

    accuracy                           0.84      4957
   macro avg       0.51      0.51      0.51      4957
weighted avg       0.78      0.84      0.80      4957


/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision and F-score
are ill-defined and being set to 0.0 in labels with no predicted
samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: UndefinedMetricWarning: Precision and F-score are ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: UndefinedMetricWarning: Precision and F-score are ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

FASTTEXT REPRESENTATION

```python
# FastText
fasttext_model = FastText(sentences=X_train, vector_size=100,
window=5, min_count=1, workers=4)
X_train_fasttext = np.array([np.mean([fasttext_model.wv[word] for word
in words if word in fasttext_model.wv] or [np.zeros(100)], axis=0) for
words in X_train])
X_test_fasttext = np.array([np.mean([fasttext_model.wv[word] for word
in words if word in fasttext_model.wv] or [np.zeros(100)], axis=0) for
words in X_test])
```

```
# Print Word Representations from FastText
print("Word Representations from FastText:")
for word, representation in zip(fasttext_model.wv.index_to_key[:10],
fasttext_model.wv.vectors[:10]):
    print(word, representation)
```

```
Word Representations from FastText:
bitch [-1.2081189   0.48140657 -0.45500606  1.2237934   1.1582677  -
0.05025363
  0.7210156   0.9125101   0.8744453  -1.0732038  -0.8917567
0.20820488
 -0.921073    1.3633486   0.27999055  0.07162543  0.4194755
0.13796261
 -0.3434904  -1.5783442  -1.1715018   0.41978264 -0.30339175
0.68986714
 -0.99670464 -0.4082996  -0.22610721 -0.2613036   1.3433273
0.05678868
 -0.5205509  -0.47839132  0.5786245  -0.20522532  0.21177533
0.7629824
  0.15956216  0.18594545 -1.0617661   1.0013833   0.47804418 -
1.0387074
 -0.5291364  -0.74315053 -0.1634434  -0.7592818  -0.9658449  -
0.5481073
  0.5796927   0.2616041   0.23131043  0.05281986  1.3860446
0.3047956
 -0.5449317  -0.17371115  0.32753363  0.9444184  -0.83536357 -0.223976
 -0.31982696 -0.36510965 -1.398876    1.8386248   0.09219341
0.9639278
 -0.11818993 -0.3380057  -0.6112552   0.91946375  0.2665039  -
0.15656362
  0.29125014 -0.46849912 -0.14190103  0.6191236   0.71286213
0.37444732
 -0.07689146  0.38812813 -0.4778784   0.05001466 -0.11150043
0.14859453
 -1.3290865  -0.4824744  -0.60997486 -0.9425098  -0.18114291 -
0.4897264
 -1.3153588   0.47450072 -0.30876562  0.05977837 -1.2673143   0.527363
  0.16075726 -0.4572433   0.05172526  0.8780069 ]
rt [-2.7496347   1.0504054  -1.3901086   2.3073483   2.4153552  -
0.31854275
  1.489082    1.8033793   2.0212789  -2.2037826  -1.8674648
0.5772581
 -1.902472    2.8324425   0.5767791  -0.00817408  0.7812214
0.29590183
 -0.686792   -3.4450388  -2.4159226   1.0323925  -0.67513853
1.3494112
 -1.9564731  -0.9062413  -0.8009977  -0.59927535  2.9215477
0.08936673
 -1.0696691  -0.98508173  1.284506   -0.478127    0.5083511
```

```
1.6097856
  0.39324898  0.50343007 -2.1686258   1.9794419   1.1741107  -
2.2033834
 -0.891938   -1.282522   -0.5479101  -1.6962113  -2.1627212  -
1.1557577
  1.0119731   0.7154243   0.773574    0.10132892  2.9366636
0.68471855
 -1.0782474  -0.5768193   0.7477412   2.0869513  -1.6230922  -
0.41734654
 -0.5501147  -0.76092196 -3.2738476   3.7451804   0.29829505
1.9607964
 -0.17441475 -0.7743221  -1.1280054   2.1135454   0.4867701  -
0.16574705
  0.27739382 -1.2157577  -0.14938614  1.4291604   1.4101024
0.5770287
 -0.10695593  0.8866128  -1.1013889   0.16684486 -0.20893484
0.2304841
 -2.9637537  -1.0951412  -1.2298493  -1.8142098  -0.38646674 -
0.8799437
 -2.7495203   0.8654108  -0.71256745  0.14464976 -2.9050283
1.1781716
  0.44013202 -0.89197636 -0.03122978  1.9781842 ]
128514 [-2.3083916   1.5123767  -0.99673533  1.5836414   2.4889445  -
1.3962185
  2.636454    0.14662112  1.4656582  -1.5649377  -1.3985373
0.7342893
 -1.8121009   2.0080445  -0.6004383  -0.45694673 -0.33014578
1.3974966
 -0.6050877  -1.3093249  -0.8782706   1.9692063  -0.59884065
3.0945022
 -1.9313599   0.19779481 -2.9146423  -2.945589    0.88009584 -
0.8353649
 -0.8305242  -0.81261116  0.52280223 -0.9090523  -0.04753865
0.77657264
  0.50665134 -1.2217375  -2.8662689   0.20158188 -0.33843076 -
0.7590947
 -0.207563   -0.78946835  2.6796525  -0.08445553 -0.39669028 -2.185235
  0.8633479  -0.22140904 -0.0063928   0.45095614  1.4020596  -
0.9630478
 -0.8786332  -0.73066854  0.68460304  2.3729317  -1.5456345  -
1.1199212
 -0.35845593 -1.3424062  -2.8368976   2.2222502   1.014693
1.0119048
  1.3841823  -0.6635896  -0.5217548   1.2408847   0.98662215
1.1625718
 -1.208719    0.26903114 -1.7319643   1.599141   -0.69472617
0.58098906
  0.10768061 -0.19006051 -2.029782   -0.57369226  0.67270833
0.2970957
```

```
 -3.2988799  -0.255656    0.73623747 -0.29487213  0.09855852 -
1.0761479
 -0.08581085  0.64296144  0.29774106 -1.8295456  -1.3921844
1.2195275
  0.02745261 -0.04562754 -0.09984064  0.5065179 ]
bitches [-1.1784014   0.45288894 -0.4792722   1.1467189   1.1102062  -
0.06598108
  0.6835858   0.8468142   0.8440738  -1.0181049  -0.8558744
0.22626394
 -0.87331754  1.2903365   0.27201793  0.05227335  0.39187822
0.13877149
 -0.32624215 -1.5029373  -1.1172812   0.41152412 -0.29290944
0.65531963
 -0.939998   -0.39140543 -0.24756536 -0.27468356  1.2773073
0.04210895
 -0.4902594  -0.45158088  0.54778826 -0.19622597  0.20591676
0.72948223
  0.16079184  0.18134722 -1.0090734   0.93436027  0.46561846 -
0.9866305
 -0.46956235 -0.68106073 -0.16148415 -0.7120761  -0.9242214  -
0.5188408
  0.5419522   0.25642025  0.23884317  0.05872887  1.3232524
0.2874746
 -0.51553714 -0.1777597   0.30607668  0.91094    -0.7853867  -
0.22561535
 -0.29948124 -0.34533545 -1.3549731   1.7247534   0.08494794
0.9019349
 -0.09820019 -0.3280598  -0.5674504   0.8882628   0.25068292 -
0.13797306
  0.23247391 -0.4513874  -0.1336621   0.5995876   0.66246516
0.35687587
 -0.06798356  0.37832227 -0.46543834  0.05475136 -0.11069231
0.13811915
 -1.2704402  -0.4525688  -0.5603188  -0.8826789  -0.17591022 -
0.46759057
 -1.2374054   0.44424874 -0.30366468  0.04808395 -1.2277983   0.510539
  0.16734976 -0.42178228  0.02991003  0.8308974 ]
http [-1.5779427   0.75316674 -0.8242067   1.1414391   1.5128303  -
0.5530266
  1.2126484   0.5805059   1.0887946  -1.1540145  -0.9887628
0.4912956
 -1.0948714   1.4501841  -0.03514142 -0.18278313  0.1180722
0.55133843
 -0.3815094  -1.4902282  -1.0334271   0.92709297 -0.42090252
1.3367294
 -1.0808644  -0.23133898 -1.2002805  -1.069816    1.2289776  -
0.23664223
 -0.5782833  -0.5689546   0.57011425 -0.3883918   0.14761628
0.73461145
```

```
  0.26984575 -0.12496364 -1.4635205   0.64800906  0.350298   -
0.9378227
 -0.30098253 -0.56882143  0.5676801  -0.60348743 -0.8328504  -
0.96478003
  0.55101186  0.21257545  0.2997877   0.20591284  1.3467206  -
0.05781426
 -0.61248654 -0.42275965  0.4807665   1.3887302  -0.9143247  -
0.45983064
 -0.27965295 -0.60143876 -1.9348778   1.7471244   0.4170639
0.8561756
  0.41234818 -0.4445669  -0.4721248   1.057228    0.4507384
0.31195432
 -0.40221015 -0.37030116 -0.49699956  0.9304659   0.24449728
0.3027785
  0.01954531  0.258319   -0.9632111  -0.11244116  0.11520541
0.13239773
 -1.8929397  -0.44817045 -0.13584174 -0.64674693 -0.12384856 -
0.5445742
 -0.8948959   0.4589183  -0.17233507 -0.48449212 -1.4104946
0.73300916
  0.18779182 -0.27061266 -0.0522537   0.8079752 ]
like [-1.2951671   0.45401257 -0.5925756   1.2389922   1.1011577
0.03573859
  0.59841686  1.0930873   0.99377036 -1.0802829  -0.93933564
0.16963488
 -0.91290575  1.4552907   0.4252758   0.06959087  0.4906331  -
0.02552666
 -0.34545738 -1.8487226  -1.3137285   0.33837566 -0.33931124  0.484776
 -1.0323142  -0.55581003 -0.04645269 -0.0277959   1.5552051
0.18467814
 -0.52681214 -0.4698713   0.6629579  -0.20226474  0.31294718
0.80996037
  0.17210865  0.41721812 -1.0221236   1.1705697   0.6711097  -
1.1594102
 -0.5612487  -0.7506933  -0.5589224  -0.9227803  -1.1981246  -
0.4709446
  0.5342052   0.3844512   0.4162284  -0.01274842  1.5349721
0.50461227
 -0.52906364 -0.21976419  0.35168037  0.9249706  -0.8051428  -
0.15746872
 -0.25751182 -0.30592534 -1.5417784   2.0379572   0.05236953
1.0742917
 -0.2916597  -0.35091424 -0.63719547  1.0439043   0.21976124 -
0.23398638
  0.4269434  -0.63978314  0.05973117  0.6328359   0.88090116
0.30236885
 -0.12218055  0.47442952 -0.38272217  0.15785284 -0.17974395
0.10316968
 -1.3505718  -0.6114059  -0.8672434  -1.0222884  -0.23829469 -
```

```
0.43474934
 -1.6115738   0.448897   -0.42274553  0.26460707 -1.4570094
0.5348459
  0.2603744  -0.5392822   0.05540726  1.0595222 ]
hoes [-1.5552268e+00  6.1658478e-01 -6.7998981e-01  1.3954655e+00
  1.4081267e+00 -1.7369141e-01  9.1758692e-01  1.0200222e+00
  1.1003633e+00 -1.2818106e+00 -1.0980437e+00  3.1309319e-01
 -1.1081836e+00  1.6280346e+00  2.9771471e-01  2.3934077e-02
  4.3492699e-01  2.2561732e-01 -4.0121070e-01 -1.9041713e+00
 -1.3653156e+00  5.7906061e-01 -3.7955007e-01  8.9684176e-01
 -1.1934726e+00 -4.6818969e-01 -4.9536487e-01 -4.4784263e-01
  1.5851295e+00  1.7670752e-02 -6.2667423e-01 -5.5899853e-01
  6.9468331e-01 -2.7653405e-01  2.6057637e-01  9.0487474e-01
  2.4138723e-01  1.7749399e-01 -1.3252305e+00  1.0980568e+00
  5.9335065e-01 -1.2228096e+00 -5.4520488e-01 -7.9306489e-01
 -1.2043849e-01 -8.7142271e-01 -1.1528610e+00 -7.0562100e-01
  6.5640795e-01  3.3738473e-01  3.4641197e-01  8.6864829e-02
  1.6425949e+00  3.1335041e-01 -6.2685782e-01 -2.9205754e-01
  4.2954740e-01  1.2222426e+00 -9.8187810e-01 -3.1351176e-01
 -3.5751945e-01 -4.5686367e-01 -1.8232703e+00  2.1413450e+00
  1.7841734e-01  1.1384953e+00 -4.4720899e-02 -4.3130931e-01
 -6.8712145e-01  1.1545117e+00  3.3630317e-01 -9.1318257e-02
  1.7860951e-01 -5.7502055e-01 -1.9869202e-01  8.1498438e-01
  7.5201112e-01  3.9488789e-01 -7.4752800e-02  4.4628403e-01
 -6.5342182e-01  4.4917282e-02 -9.8001011e-02  1.4827232e-01
 -1.7168919e+00 -5.8638859e-01 -6.4600325e-01 -1.0391577e+00
 -2.1427654e-01 -5.6861204e-01 -1.4881607e+00  5.4281849e-01
 -3.7171009e-01 -3.2269575e-03 -1.5647565e+00  6.7322564e-01
  2.2780477e-01 -5.1134300e-01 -3.7303296e-04  1.0446757e+00]
pussy [-1.2647748   0.48905516 -0.6015518   1.1212136   1.1494861  -
0.13072987
  0.7193665   0.86087596  0.92548907 -1.0511653  -0.88384175
0.25218788
 -0.899747    1.3276619   0.25729245  0.00781082  0.36821356
0.15226682
 -0.3386096  -1.5844852  -1.1336762   0.48371944 -0.32550663
0.68090194
 -0.93323    -0.41366285 -0.37207556 -0.30861837  1.3557947
0.0266632
 -0.5045611  -0.4734678   0.5922269  -0.22095022  0.2210234
0.74632794
  0.17782554  0.19828299 -1.054239    0.92870724  0.54074085 -1.020069
 -0.4456648  -0.6323488  -0.20080282 -0.7798518  -0.9914959  -0.566356
  0.5039761   0.30173698  0.32306328  0.06875513  1.3746516   0.306725
 -0.5169105  -0.24760462  0.34280366  0.9911596  -0.78053933 -
0.21932611
 -0.25902054 -0.36363512 -1.4994284   1.7809275   0.14024092
0.9234922
 -0.06705476 -0.35064882 -0.5320324   0.96752036  0.25812694 -
```

```
0.07788887
  0.16361135 -0.5171877  -0.12790236  0.66214675  0.6629007
0.28523192
 -0.06605412  0.4054552  -0.52479327  0.05740597 -0.10085697
0.12587883
 -1.379649   -0.49146944 -0.5716406  -0.8555216  -0.18339483 -
0.43282714
 -1.284647    0.42367142 -0.32212278  0.04984748 -1.3359141
0.55220973
  0.20042215 -0.43127742 -0.00499741  0.8962367 ]
hoe [-1.5362898   0.6266328  -0.66223246  1.4016775   1.4059986  -
0.17102234
  0.9263167   1.0176674   1.1015614  -1.2751504  -1.0854738
0.2822512
 -1.1074007   1.6237289   0.28626886  0.02104355  0.433239
0.20553613
 -0.39775148 -1.897892   -1.3607914   0.58300495 -0.379821
0.89448196
 -1.1980524  -0.47179025 -0.4802483  -0.4340187   1.5774562
0.01900415
 -0.6179494  -0.5691337   0.69985384 -0.28196687  0.2583473   0.897096
  0.22439477  0.17818533 -1.320527    1.1070249   0.5774723  -
1.2204918
 -0.56448406 -0.8028162  -0.12082542 -0.8898831  -1.1548072  -
0.7194442
  0.65480596  0.33629224  0.34276167  0.07271296  1.6414979
0.31722757
 -0.6163263  -0.3061655   0.437282    1.2190675  -0.9872698  -
0.30178523
 -0.34231636 -0.46782544 -1.8196675   2.1634243   0.19224285
1.1397812
 -0.05829192 -0.4292411  -0.676689    1.1440412   0.33156443 -
0.08942544
  0.20733842 -0.5793017  -0.19638765  0.81077844  0.74863356
0.38655034
 -0.0768723   0.44776413 -0.657252    0.03894938 -0.0803957
0.15078786
 -1.714241   -0.585022   -0.6687516  -1.0273442  -0.20683588 -
0.55298424
 -1.4894692   0.53276575 -0.36055636 -0.00624196 -1.552138
0.6728969
  0.21594416 -0.50613314  0.01788813  1.0484201 ]
8220 [-1.8396721   0.88307667 -0.9033182   1.3938209   1.7621266  -
0.5938236
  1.4503815   0.69894576  1.2157696  -1.3632603  -1.1592793
0.5101465
 -1.3308578   1.7360743  -0.01799154 -0.17011608  0.16313428
0.62486064
 -0.45466483 -1.7050934  -1.194971    1.073434   -0.4844821
```

```
1.5852181
 -1.3351907  -0.25061938 -1.3418839  -1.2631706   1.4211128  -
0.26711848
 -0.6709707  -0.653378    0.6573543  -0.48323354  0.17514578  0.87002
  0.32420957 -0.19968963 -1.776615    0.78351694  0.37277898 -
1.0747669
 -0.38786316 -0.7037051   0.70376515 -0.6294375  -0.9491071  -
1.1426286
  0.6819098   0.21766096  0.30055696  0.22097078  1.5749269  -
0.07966774
 -0.7106212  -0.43938467  0.513474    1.6041442  -1.1092991  -
0.53786826
 -0.34235343 -0.7170678  -2.192793    2.0986962   0.46422356
1.0475676
  0.4504614  -0.49573967 -0.57755136  1.196844    0.5411582
0.33845857
 -0.38481054 -0.35369387 -0.6194159   1.0579509   0.2902908   0.40529
  0.01583681  0.27794793 -1.1107337  -0.13812615  0.15018873
0.1831263
 -2.2163699  -0.4825361  -0.17918772 -0.7710337  -0.14228763 -
0.68254006
 -1.039643    0.57064617 -0.20420438 -0.5620372  -1.6040212
0.84744877
  0.20611529 -0.3399194  -0.05801708  0.89089227]

# Evaluate FastText + Logistic Regression
y_pred_fasttext = logreg_fasttext.predict(X_test_fasttext)
fasttext_accuracy = accuracy_score(y_test, y_pred_fasttext)
print("FastText + Logistic Regression Accuracy:", fasttext_accuracy)

# Analyze misclassified instances for FastText + Logistic Regression
misclassified_fasttext = X_test[y_test != y_pred_fasttext]
true_labels_fasttext = y_test[y_test != y_pred_fasttext]
predicted_labels_fasttext = y_pred_fasttext[y_test != y_pred_fasttext]
misclassified_df_fasttext = pd.DataFrame({'Text':
misclassified_fasttext, 'True Label': true_labels_fasttext, 'Predicted
Label': predicted_labels_fasttext})
print("Misclassified instances for FastText + Logistic Regression:")
print(misclassified_df_fasttext)

# Generate confusion matrix and classification report for FastText +
Logistic Regression
print("Confusion Matrix for FastText + Logistic Regression:")
print(confusion_matrix(y_test, y_pred_fasttext))
print("\nClassification Report for FastText + Logistic Regression:")
print(classification_report(y_test, y_pred_fasttext))

FastText + Logistic Regression Accuracy: 0.8355860399435142
Misclassified instances for FastText + Logistic Regression:
                                                    Text  True
```

```
Label  \
18943  [rt, 1inkkofrosess, lol, credit, ai, near, goo...          2

4273             [search, gay, redneck, episode, 1, play]          0

3778   [keebithalal, loganswarning, got, ta, love, is...          0

15789  [rt, jsu, coach, omar, johnson, u, ball, u, th...          2

11311  [tryna, get, sleep, birds, start, getting, rowdy]          2

...                                                  ...        ...

10959                         [think, eat, brownie, pass]          2

20979  [real, unreal, lol, yankees, worldseries, 27an...          2

7339                        [xcorey21, uh, trash, 128536]          1

20769  [unfollowed, said, cried, watching, dawn, apes...          2

3310   [grizzboadams, wyattnuckels, haha, ight, nig, ...          0


       Predicted Label
18943                1
4273                 2
3778                 1
15789                1
11311                1
...                ...
10959                1
20979                1
7339                 2
20769                1
3310                 1

[815 rows x 3 columns]
Confusion Matrix for FastText + Logistic Regression:
[[   0  219   71]
 [   0 3685  147]
 [   0  378  457]]

Classification Report for FastText + Logistic Regression:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       290
           1       0.86      0.96      0.91      3832
           2       0.68      0.55      0.61       835

    accuracy                           0.84      4957
```

```
   macro avg       0.51      0.50      0.50      4957
weighted avg       0.78      0.84      0.80      4957
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision and F-score
are ill-defined and being set to 0.0 in labels with no predicted
samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: UndefinedMetricWarning: Precision and F-score are ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: UndefinedMetricWarning: Precision and F-score are ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

CNN AND RNN

```
# CNN
tokenizer = Tokenizer()
tokenizer.fit_on_texts(X_train)
X_train_cnn = tokenizer.texts_to_sequences(X_train)
X_test_cnn = tokenizer.texts_to_sequences(X_test)
vocab_size = len(tokenizer.word_index) + 1
maxlen = 100
X_train_cnn = pad_sequences(X_train_cnn, padding='post',
maxlen=maxlen)
X_test_cnn = pad_sequences(X_test_cnn, padding='post', maxlen=maxlen)


# RNN
X_train_rnn = pad_sequences(X_train_cnn, padding='post',
maxlen=maxlen)
X_test_rnn = pad_sequences(X_test_cnn, padding='post', maxlen=maxlen)


# Define CNN model
cnn_model = Sequential()
cnn_model.add(Embedding(input_dim=vocab_size, output_dim=100,
input_length=maxlen))
cnn_model.add(Conv1D(filters=64, kernel_size=3, activation='relu'))
cnn_model.add(MaxPooling1D(pool_size=2))
cnn_model.add(Dense(10, activation='relu'))
cnn_model.add(Dense(1, activation='sigmoid'))
cnn_model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])
```

```
# Train CNN model
cnn_model.fit(X_train_cnn, y_train, epochs=10, batch_size=64,
validation_data=(X_test_cnn, y_test))

Epoch 1/10
310/310 [==============================] - 29s 91ms/step - loss: -
1.3062 - accuracy: 0.7724 - val_loss: -1.6817 - val_accuracy: 0.7731
Epoch 2/10
310/310 [==============================] - 23s 74ms/step - loss: -
1.6998 - accuracy: 0.7746 - val_loss: -1.6891 - val_accuracy: 0.7730
Epoch 3/10
310/310 [==============================] - 23s 75ms/step - loss: -
1.7143 - accuracy: 0.7746 - val_loss: -1.6899 - val_accuracy: 0.7729
Epoch 4/10
310/310 [==============================] - 22s 72ms/step - loss: -
1.7203 - accuracy: 0.7749 - val_loss: -1.6907 - val_accuracy: 0.7730
Epoch 5/10
310/310 [==============================] - 22s 70ms/step - loss: -
1.7245 - accuracy: 0.7754 - val_loss: -1.6921 - val_accuracy: 0.7729
Epoch 6/10
310/310 [==============================] - 20s 64ms/step - loss: -
1.7285 - accuracy: 0.7757 - val_loss: -1.6940 - val_accuracy: 0.7729
Epoch 7/10
310/310 [==============================] - 19s 62ms/step - loss: -
1.7328 - accuracy: 0.7761 - val_loss: -1.6946 - val_accuracy: 0.7726
Epoch 8/10
310/310 [==============================] - 20s 66ms/step - loss: -
1.7355 - accuracy: 0.7765 - val_loss: -1.6934 - val_accuracy: 0.7725
Epoch 9/10
310/310 [==============================] - 24s 79ms/step - loss: -
1.7380 - accuracy: 0.7766 - val_loss: -1.6916 - val_accuracy: 0.7721
Epoch 10/10
310/310 [==============================] - 24s 78ms/step - loss: -
1.7404 - accuracy: 0.7769 - val_loss: -1.6921 - val_accuracy: 0.7721

<keras.src.callbacks.History at 0x7c326e32feb0>


# Define RNN model
rnn_model = Sequential()
rnn_model.add(Embedding(input_dim=vocab_size, output_dim=100,
input_length=maxlen))
rnn_model.add(LSTM(100))
rnn_model.add(Dense(1, activation='sigmoid'))
rnn_model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])

# Train RNN model
```

```python
rnn_model.fit(X_train_rnn, y_train, epochs=10, batch_size=64,
validation_data=(X_test_rnn, y_test))
```

```
Epoch 1/10
310/310 [==============================] - 71s 220ms/step - loss: -
2.6451 - accuracy: 0.7746 - val_loss: -4.2537 - val_accuracy: 0.7730
Epoch 2/10
310/310 [==============================] - 69s 222ms/step - loss: -
5.7099 - accuracy: 0.7746 - val_loss: -7.1416 - val_accuracy: 0.7730
Epoch 3/10
310/310 [==============================] - 67s 216ms/step - loss: -
8.5995 - accuracy: 0.7746 - val_loss: -10.0082 - val_accuracy: 0.7730
Epoch 4/10
310/310 [==============================] - 64s 207ms/step - loss: -
11.4720 - accuracy: 0.7746 - val_loss: -12.8580 - val_accuracy: 0.7730
Epoch 5/10
310/310 [==============================] - 66s 214ms/step - loss: -
14.3444 - accuracy: 0.7746 - val_loss: -15.7205 - val_accuracy: 0.7730
Epoch 6/10
310/310 [==============================] - 68s 220ms/step - loss: -
17.1988 - accuracy: 0.7746 - val_loss: -18.5489 - val_accuracy: 0.7730
Epoch 7/10
310/310 [==============================] - 67s 215ms/step - loss: -
20.0383 - accuracy: 0.7746 - val_loss: -21.3850 - val_accuracy: 0.7730
Epoch 8/10
310/310 [==============================] - 66s 213ms/step - loss: -
22.8740 - accuracy: 0.7746 - val_loss: -24.1974 - val_accuracy: 0.7730
Epoch 9/10
310/310 [==============================] - 69s 224ms/step - loss: -
25.7058 - accuracy: 0.7746 - val_loss: -27.0339 - val_accuracy: 0.7730
Epoch 10/10
310/310 [==============================] - 65s 209ms/step - loss: -
28.5416 - accuracy: 0.7746 - val_loss: -29.8564 - val_accuracy: 0.7730

<keras.src.callbacks.History at 0x7c326884ca00>
```

```python
# Evaluate CNN
cnn_loss, cnn_accuracy = cnn_model.evaluate(X_test_cnn, y_test)
print("CNN Accuracy:", cnn_accuracy)

# Evaluate RNN
rnn_loss, rnn_accuracy = rnn_model.evaluate(X_test_rnn, y_test)
print("RNN Accuracy:", rnn_accuracy)
```

```
155/155 [==============================] - 1s 8ms/step - loss: -1.6921
- accuracy: 0.7721
CNN Accuracy: 0.7720804214477539
155/155 [==============================] - 6s 41ms/step - loss: -
```

```
29.8564 - accuracy: 0.7730
RNN Accuracy: 0.7730482220649719
```

```python
# Get Embedding Layer Output for CNN
embedding_output_cnn = cnn_model.layers[0](X_test_cnn)
print("Embedding Output Shape (CNN):", embedding_output_cnn.shape)
```

```
Embedding Output Shape (CNN): (4957, 100, 100)
```

```python
# Get Embedding Layer Output for RNN
embedding_output_rnn = rnn_model.layers[0](X_test_rnn)
print("Embedding Output Shape (RNN):", embedding_output_rnn.shape)
```

```
Embedding Output Shape (RNN): (4957, 100, 100)
```