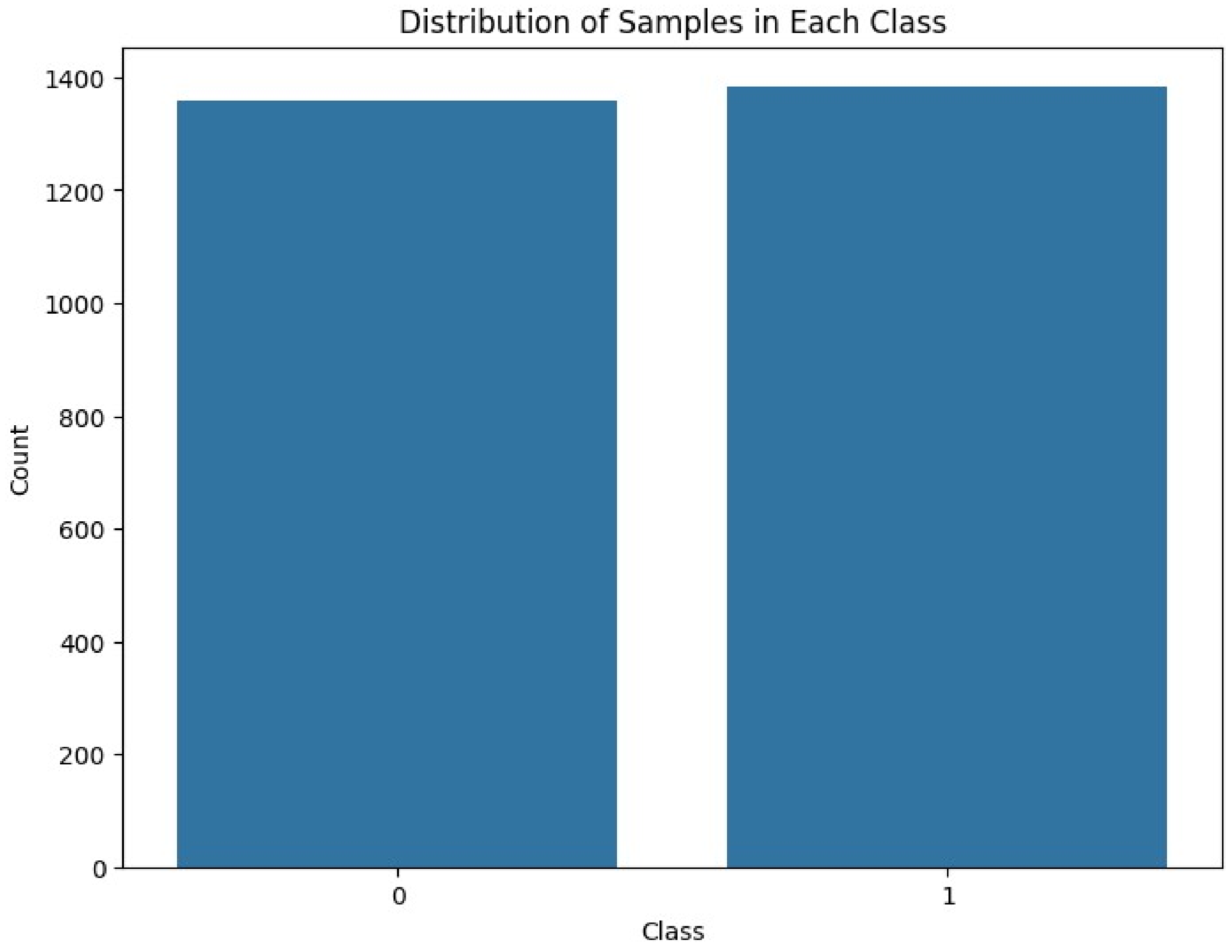
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
import seaborn as sns
import spacy
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.svm import LinearSVC
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import LabelEncoder
from tensorflow.keras.models import Sequential, save_model, load_model
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping
from joblib import dump
from joblib import load

# load data
data_pro = pd.read_csv('sentimentdataset (Project 1).csv')
# print the first rows
print(data_pro.head())
print()

# print describtion
print(data_pro.describe())
print()

# distribution of samples in each class
distribution = data_pro['Target'].value_counts()
print(distribution)
print()

# plot distribution of samples in each class
plt.figure(figsize=(8, 6))
sns.countplot(x='Target', data=data_pro)
plt.title('Distribution of Samples in Each Class')
plt.xlabel('Class')
plt.ylabel('Count')
plt.show()
```

```
  Source  ID                                            Message
Target
0   Yelp   0                                 Crust is not good.
0
1   Yelp   1           Not tasty and the texture was just nasty.
0
2   Yelp   2  Stopped by during the late May bank holiday of...
```

```
1
3   Yelp   3  The selection on the menu was great and so wer...
1
4   Yelp   4     Now I am getting angry and I want my damn pho.
0

                ID       Target
count  2745.000000  2745.000000
mean    464.711475     0.504554
std     276.335259     0.500070
min       0.000000     0.000000
25%     228.000000     0.000000
50%     457.000000     1.000000
75%     686.000000     1.000000
max     998.000000     1.000000

Target
1    1385
0    1360
Name: count, dtype: int64
```




```
# drop id and source columns
data = data_pro.drop(columns=['Source', 'ID'])
print(data)
print()
```

```
                                                Message  Target
0                                    Crust is not good.       0
1             Not tasty and the texture was just nasty.       0
2     Stopped by during the late May bank holiday of...       1
3     The selection on the menu was great and so wer...       1
4        Now I am getting angry and I want my damn pho.       0
...                                                 ...     ...
2740  The screen does get smudged easily because it ...       0
2741  What a piece of junk.. I lose more calls on th...       0
2742                         Item Does Not Match Picture.       0
2743  The only thing that disappoint me is the infra...       0
2744  You can not answer calls with the unit, never ...       0

[2745 rows x 2 columns]
```

```python
# spacy.cli.download("en_core_web_sm")
lemmatization = spacy.load('en_core_web_sm')

# eliminate stop words, perform lemmatization for each sentiment
def perform_lemmatization(all_messages):
    all_messages = lemmatization(all_messages)
    new_sentiment = [sentiment.lemma_ for sentiment in all_messages if
not sentiment.is_stop]
    return " ".join(new_sentiment)

data['Message'] = data['Message'].apply(perform_lemmatization)

print(data)
print()
```

```
                                                Message  Target
0                                           crust good .       0
1                                   tasty texture nasty .       0
2     stop late bank holiday Rick Steve recommendati...       1
3                            selection menu great price .       1
4                               get angry want damn pho .       0
...                                                  ...     ...
2740              screen smudge easily touch ear face .       0
2741                    piece junk .. lose call phone .       0
2742                               item match Picture .       0
2743         thing disappoint infra red port ( irda ) .       0
2744                             answer call unit , work !       0

[2745 rows x 2 columns]
```

```python
# Generate sentence embeddings using CountVectorizer
count_vectorizer = CountVectorizer()
embeddings = count_vectorizer.fit_transform(data['Message'])
print(embeddings.toarray())
print()
```

```
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
```

```python
# Split into training and testing sets (75 train, 25 test)
X = embeddings
y = data['Target']
X_train, X_test, y_train, y_test =
train_test_split(X,y,test_size=0.25,random_state=42)
```

```python
print("X_train")
print(X_train.toarray())
print("----------------------------------------------")
print("X_test")
print(X_test.toarray())
print("----------------------------------------------")
print("y_train")
print(y_train)
print("----------------------------------------------")
print("y_test")
print(y_test)
```

```
X_train
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
----------------------------------------------
X_test
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
----------------------------------------------
y_train
1288    0
930     0
657     1
1620    1
1640    0
       ..
1638    0
1095    0
1130    0
1294    1
860     1
Name: Target, Length: 2058, dtype: int64
----------------------------------------------
y_test
471     1
1025    1
767     1
1896    0
430     1
```

```
       ..
1949    0
1937    1
2353    0
1057    1
2092    1
Name: Target, Length: 687, dtype: int64
```

```
# Initial Experiment
# use LinearSVC
linear_svc = LinearSVC(dual=False)
# use grid search to identify optimal parameters from this values
[0.01,0.1, 1, 10]
# cross validation is 3
gridSearchCV = GridSearchCV(linear_svc, {'C': [0.01,0.1,0.5,1]}, cv=3,
scoring='accuracy', verbose=1)
gridSearchCV.fit(X_train, y_train)

# Optimal Parameter
print("Optimal Parameter: ", gridSearchCV.best_params_)

# predict test data
y_pred = gridSearchCV.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print("Highest Accuracy: {:.2f}%".format(accuracy * 100))

print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))

# classification report for initial experiment
print("Classification Report for Initial Experiment:")
print(classification_report(y_test, y_pred))
```

```
Fitting 3 folds for each of 4 candidates, totalling 12 fits
Optimal Parameter:  {'C': 0.1}
Highest Accuracy: 79.33%
Confusion Matrix:
[[277  54]
 [ 88 268]]
Classification Report for Initial Experiment:
              precision    recall  f1-score   support

           0       0.76      0.84      0.80       331
           1       0.83      0.75      0.79       356

    accuracy                           0.79       687
   macro avg       0.80      0.79      0.79       687
weighted avg       0.80      0.79      0.79       687
```

```python
# Save the best model of SVC
bestModel = gridSearchCV.best_estimator_
dump(bestModel, 'bestModel.joblib')

# reload
loadedModel = load('bestModel.joblib')

new_data = pd.read_csv('newdata.csv')
print(new_data)
print()

new_data['Message'] = new_data['Message'].apply(perform_lemmatization)
print(new_data)
print()

embeddings1 = count_vectorizer.transform(new_data['Message'])
# predict
y_pred = loadedModel.predict(embeddings1)
print('y_pred',y_pred)
```

```
                                             Message
0                The weather outside is not pleasant.
1                                    Team is amazing
2  This novel is truly captivating, a literary ma...
3  The cityscape at night gives life to the sayin...
4  The presentation was monotonous. The guest spe...
5  The city park is unremarkable. The botanical g...
6  The smartphone's battery life is disappointing...
7  The homework assignment is tedious. The extra ...
8                   The homework assignment is easy
9  The sunrise this morning was absolutely breath...

                                             Message
0                         weather outside pleasant .
1                                       team amazing
2   novel truly captivating , literary masterpiece .
3  cityscape night give life saying , " city slee...
4  presentation monotonous . guest speaker , , de...
5  city park unremarkable . botanical garden near...
6  smartphone battery life disappointing . late m...
7  homework assignment tedious . extra credit pro...
8                          homework assignment easy
9  sunrise morning absolutely breathtaking , pain...

y_pred [0 1 1 0 0 1 0 0 1 0]
```

```python
# Subsequent Experiment
highestAccuracy = 0
optimalHyperparameters = {}
```

```
# Hyperparameter
num_neurons = [64, 128, 256]
num_learning_rate = [0.001, 0.01, 0.1]
num_batch_size = [32, 64, 128]

# Explore different hyperparameters
for neurons in num_neurons:
    for learningRate in num_learning_rate:
        for batchSize in num_batch_size:

            # ANN model
            model = Sequential()
            model.add(Dense(neurons, input_dim=X_train.shape[1],
activation='relu'))
            model.add(Dropout(0.5))
            model.add(Dense(1, activation='sigmoid'))
            optimizer = Adam(learning_rate=learningRate)
            model.compile(loss='binary_crossentropy',
optimizer=optimizer, metrics=['accuracy'])

            # fit the model
            history = model.fit(X_train.toarray(), y_train, epochs=20,
batch_size=batchSize,
                                validation_split=0.1,
callbacks=[EarlyStopping(patience=3)])

            # calculate accuracy
            loss, accuracy = model.evaluate(X_test.toarray(), y_test)
            # predict test data
            # probabilities values
            y_p = model.predict(X_test.toarray())
            # convert probabilities values to 0 or 1
            y_pred = [1 if p > 0.5 else 0 for p in y_p]
            # classification report
            classificationReport = classification_report(y_test,
y_pred)

            # compare accuracies to find the highest accuracy
            if accuracy > highestAccuracy:
                highestAccuracy = accuracy
                # optimal neuron, learning rate, batch size
                optimalHyperparameters = {'neurons': neurons,
'learningRate': learningRate, 'batchSize': batchSize}
                # optimal classification report
                optimalClassificationReport = classificationReport
                # Save the best model
                model.save('bestModelANN.keras')

print()
print("Optimal Hyperparameters:", optimalHyperparameters)
```

```python
print("Highest Accuracy: {:.2f}%".format(highestAccuracy * 100))
print("Classification Report for Best Model:")
print(optimalClassificationReport)

# Reload
loadedModelANN = load_model('bestModelANN.keras')

new_data = pd.read_csv('newdata.csv')
print(new_data)
print()

new_data['Message'] = new_data['Message'].apply(perform_lemmatization)
print(new_data)
print()

embeddings1 = count_vectorizer.transform(new_data['Message'])

y_p = loadedModelANN.predict(embeddings1.toarray())
# convert probabilities values to 0 or 1
y_pred = [1 if p > 0.5 else 0 for p in y_p]

print('y_pred',y_pred)
print()
```

```
Epoch 1/20
58/58 [==============================] - 1s 5ms/step - loss: 0.6828 -
accuracy: 0.5934 - val_loss: 0.6696 - val_accuracy: 0.7427
Epoch 2/20
58/58 [==============================] - 0s 3ms/step - loss: 0.6160 -
accuracy: 0.8261 - val_loss: 0.6136 - val_accuracy: 0.7427
Epoch 3/20
58/58 [==============================] - 0s 3ms/step - loss: 0.4910 -
accuracy: 0.8720 - val_loss: 0.5534 - val_accuracy: 0.7476
Epoch 4/20
58/58 [==============================] - 0s 3ms/step - loss: 0.3706 -
accuracy: 0.9001 - val_loss: 0.5151 - val_accuracy: 0.7670
Epoch 5/20
58/58 [==============================] - 0s 3ms/step - loss: 0.2884 -
accuracy: 0.9244 - val_loss: 0.4943 - val_accuracy: 0.7767
Epoch 6/20
58/58 [==============================] - 0s 3ms/step - loss: 0.2312 -
accuracy: 0.9374 - val_loss: 0.4920 - val_accuracy: 0.7718
Epoch 7/20
58/58 [==============================] - 0s 3ms/step - loss: 0.1971 -
accuracy: 0.9422 - val_loss: 0.4932 - val_accuracy: 0.7864
Epoch 8/20
58/58 [==============================] - 0s 3ms/step - loss: 0.1650 -
accuracy: 0.9541 - val_loss: 0.5051 - val_accuracy: 0.7816
Epoch 9/20
58/58 [==============================] - 0s 5ms/step - loss: 0.1466 -
```

```
accuracy: 0.9552 - val_loss: 0.5063 - val_accuracy: 0.7864
22/22 [==============================] - 0s 2ms/step - loss: 0.5853 -
accuracy: 0.8035
22/22 [==============================] - 0s 1ms/step
Epoch 1/20
29/29 [==============================] - 1s 9ms/step - loss: 0.6829 -
accuracy: 0.5659 - val_loss: 0.6712 - val_accuracy: 0.6893
Epoch 2/20
29/29 [==============================] - 0s 5ms/step - loss: 0.6339 -
accuracy: 0.8035 - val_loss: 0.6417 - val_accuracy: 0.7136
Epoch 3/20
29/29 [==============================] - 0s 4ms/step - loss: 0.5634 -
accuracy: 0.8612 - val_loss: 0.5993 - val_accuracy: 0.7476
Epoch 4/20
29/29 [==============================] - 0s 4ms/step - loss: 0.4780 -
accuracy: 0.8942 - val_loss: 0.5628 - val_accuracy: 0.7573
Epoch 5/20
29/29 [==============================] - 0s 5ms/step - loss: 0.3953 -
accuracy: 0.9136 - val_loss: 0.5322 - val_accuracy: 0.7670
Epoch 6/20
29/29 [==============================] - 0s 6ms/step - loss: 0.3307 -
accuracy: 0.9249 - val_loss: 0.5151 - val_accuracy: 0.7816
Epoch 7/20
29/29 [==============================] - 0s 5ms/step - loss: 0.2778 -
accuracy: 0.9390 - val_loss: 0.5016 - val_accuracy: 0.7718
Epoch 8/20
29/29 [==============================] - 0s 4ms/step - loss: 0.2458 -
accuracy: 0.9330 - val_loss: 0.4972 - val_accuracy: 0.7767
Epoch 9/20
29/29 [==============================] - 0s 6ms/step - loss: 0.2130 -
accuracy: 0.9460 - val_loss: 0.4983 - val_accuracy: 0.7864
Epoch 10/20
29/29 [==============================] - 0s 5ms/step - loss: 0.1897 -
accuracy: 0.9519 - val_loss: 0.5002 - val_accuracy: 0.7913
Epoch 11/20
29/29 [==============================] - 0s 6ms/step - loss: 0.1688 -
accuracy: 0.9579 - val_loss: 0.5011 - val_accuracy: 0.7816
22/22 [==============================] - 0s 2ms/step - loss: 0.5461 -
accuracy: 0.8020
22/22 [==============================] - 0s 2ms/step
Epoch 1/20
15/15 [==============================] - 1s 13ms/step - loss: 0.6903 -
accuracy: 0.5486 - val_loss: 0.6850 - val_accuracy: 0.6311
Epoch 2/20
15/15 [==============================] - 0s 6ms/step - loss: 0.6624 -
accuracy: 0.7732 - val_loss: 0.6711 - val_accuracy: 0.7282
Epoch 3/20
15/15 [==============================] - 0s 6ms/step - loss: 0.6271 -
accuracy: 0.8623 - val_loss: 0.6509 - val_accuracy: 0.7427
```

```
Epoch 4/20
15/15 [==============================] - 0s 6ms/step - loss: 0.5803 -
accuracy: 0.8834 - val_loss: 0.6221 - val_accuracy: 0.7573
Epoch 5/20
15/15 [==============================] - 0s 6ms/step - loss: 0.5237 -
accuracy: 0.9023 - val_loss: 0.5922 - val_accuracy: 0.7573
Epoch 6/20
15/15 [==============================] - 0s 6ms/step - loss: 0.4635 -
accuracy: 0.9050 - val_loss: 0.5638 - val_accuracy: 0.7670
Epoch 7/20
15/15 [==============================] - 0s 6ms/step - loss: 0.4071 -
accuracy: 0.9141 - val_loss: 0.5404 - val_accuracy: 0.7718
Epoch 8/20
15/15 [==============================] - 0s 5ms/step - loss: 0.3577 -
accuracy: 0.9293 - val_loss: 0.5229 - val_accuracy: 0.7864
Epoch 9/20
15/15 [==============================] - 0s 7ms/step - loss: 0.3145 -
accuracy: 0.9309 - val_loss: 0.5075 - val_accuracy: 0.7816
Epoch 10/20
15/15 [==============================] - 0s 6ms/step - loss: 0.2806 -
accuracy: 0.9368 - val_loss: 0.4965 - val_accuracy: 0.7816
Epoch 11/20
15/15 [==============================] - 0s 7ms/step - loss: 0.2539 -
accuracy: 0.9390 - val_loss: 0.4921 - val_accuracy: 0.7864
Epoch 12/20
15/15 [==============================] - 0s 7ms/step - loss: 0.2330 -
accuracy: 0.9428 - val_loss: 0.4893 - val_accuracy: 0.7767
Epoch 13/20
15/15 [==============================] - 0s 7ms/step - loss: 0.2092 -
accuracy: 0.9492 - val_loss: 0.4877 - val_accuracy: 0.7913
Epoch 14/20
15/15 [==============================] - 0s 6ms/step - loss: 0.1925 -
accuracy: 0.9546 - val_loss: 0.4855 - val_accuracy: 0.7913
Epoch 15/20
15/15 [==============================] - 0s 6ms/step - loss: 0.1814 -
accuracy: 0.9552 - val_loss: 0.4866 - val_accuracy: 0.7816
Epoch 16/20
15/15 [==============================] - 0s 5ms/step - loss: 0.1648 -
accuracy: 0.9579 - val_loss: 0.4889 - val_accuracy: 0.7816
Epoch 17/20
15/15 [==============================] - 0s 5ms/step - loss: 0.1613 -
accuracy: 0.9546 - val_loss: 0.4889 - val_accuracy: 0.7718
22/22 [==============================] - 0s 2ms/step - loss: 0.5250 -
accuracy: 0.7991
22/22 [==============================] - 0s 1ms/step
Epoch 1/20
58/58 [==============================] - 1s 5ms/step - loss: 0.5776 -
accuracy: 0.7036 - val_loss: 0.5442 - val_accuracy: 0.7718
Epoch 2/20
```

```
58/58 [==============================] - 0s 3ms/step - loss: 0.2686 -
accuracy: 0.8850 - val_loss: 0.5495 - val_accuracy: 0.7816
Epoch 3/20
58/58 [==============================] - 0s 3ms/step - loss: 0.1731 -
accuracy: 0.9401 - val_loss: 0.6233 - val_accuracy: 0.7718
Epoch 4/20
58/58 [==============================] - 0s 3ms/step - loss: 0.1073 -
accuracy: 0.9584 - val_loss: 0.7099 - val_accuracy: 0.7816
22/22 [==============================] - 0s 2ms/step - loss: 0.8776 -
accuracy: 0.7831
22/22 [==============================] - 0s 1ms/step
Epoch 1/20
29/29 [==============================] - 1s 8ms/step - loss: 0.5943 -
accuracy: 0.7003 - val_loss: 0.5221 - val_accuracy: 0.7476
Epoch 2/20
29/29 [==============================] - 0s 4ms/step - loss: 0.2902 -
accuracy: 0.8871 - val_loss: 0.5388 - val_accuracy: 0.7767
Epoch 3/20
29/29 [==============================] - 0s 4ms/step - loss: 0.1840 -
accuracy: 0.9368 - val_loss: 0.6589 - val_accuracy: 0.8010
Epoch 4/20
29/29 [==============================] - 0s 4ms/step - loss: 0.1201 -
accuracy: 0.9541 - val_loss: 0.7082 - val_accuracy: 0.7816
22/22 [==============================] - 0s 1ms/step - loss: 0.9217 -
accuracy: 0.7802
22/22 [==============================] - 0s 1ms/step
Epoch 1/20
15/15 [==============================] - 1s 14ms/step - loss: 0.6299 -
accuracy: 0.6841 - val_loss: 0.5440 - val_accuracy: 0.7427
Epoch 2/20
15/15 [==============================] - 0s 6ms/step - loss: 0.3393 -
accuracy: 0.8823 - val_loss: 0.5195 - val_accuracy: 0.7767
Epoch 3/20
15/15 [==============================] - 0s 6ms/step - loss: 0.1879 -
accuracy: 0.9293 - val_loss: 0.5423 - val_accuracy: 0.7961
Epoch 4/20
15/15 [==============================] - 0s 9ms/step - loss: 0.1274 -
accuracy: 0.9541 - val_loss: 0.6227 - val_accuracy: 0.7961
Epoch 5/20
15/15 [==============================] - 0s 10ms/step - loss: 0.0933 -
accuracy: 0.9649 - val_loss: 0.6612 - val_accuracy: 0.7864
22/22 [==============================] - 0s 1ms/step - loss: 0.8276 -
accuracy: 0.7875
22/22 [==============================] - 0s 1ms/step
Epoch 1/20
58/58 [==============================] - 1s 5ms/step - loss: 0.6115 -
accuracy: 0.7046 - val_loss: 0.5212 - val_accuracy: 0.7670
Epoch 2/20
58/58 [==============================] - 0s 3ms/step - loss: 0.5673 -
```

```
accuracy: 0.8175 - val_loss: 1.0754 - val_accuracy: 0.7039
Epoch 3/20
58/58 [==============================] - 0s 3ms/step - loss: 0.3570 -
accuracy: 0.8726 - val_loss: 0.9396 - val_accuracy: 0.7621
Epoch 4/20
58/58 [==============================] - 0s 3ms/step - loss: 0.2902 -
accuracy: 0.9012 - val_loss: 0.9436 - val_accuracy: 0.7913
22/22 [==============================] - 0s 1ms/step - loss: 1.4774 -
accuracy: 0.7744
22/22 [==============================] - 0s 1ms/step
Epoch 1/20
29/29 [==============================] - 1s 8ms/step - loss: 0.5945 -
accuracy: 0.6955 - val_loss: 0.5590 - val_accuracy: 0.7718
Epoch 2/20
29/29 [==============================] - 0s 4ms/step - loss: 0.3158 -
accuracy: 0.8737 - val_loss: 0.7719 - val_accuracy: 0.7670
Epoch 3/20
29/29 [==============================] - 0s 4ms/step - loss: 0.2005 -
accuracy: 0.9260 - val_loss: 0.8492 - val_accuracy: 0.7864
Epoch 4/20
29/29 [==============================] - 0s 4ms/step - loss: 0.1566 -
accuracy: 0.9379 - val_loss: 0.8896 - val_accuracy: 0.7961
22/22 [==============================] - 0s 1ms/step - loss: 1.2230 -
accuracy: 0.7904
22/22 [==============================] - 0s 1ms/step
Epoch 1/20
15/15 [==============================] - 1s 13ms/step - loss: 0.5824 -
accuracy: 0.6971 - val_loss: 0.5991 - val_accuracy: 0.7136
Epoch 2/20
15/15 [==============================] - 0s 5ms/step - loss: 0.3132 -
accuracy: 0.8796 - val_loss: 0.7965 - val_accuracy: 0.7767
Epoch 3/20
15/15 [==============================] - 0s 5ms/step - loss: 0.1734 -
accuracy: 0.9357 - val_loss: 0.8427 - val_accuracy: 0.7816
Epoch 4/20
15/15 [==============================] - 0s 5ms/step - loss: 0.1243 -
accuracy: 0.9487 - val_loss: 0.8224 - val_accuracy: 0.8010
22/22 [==============================] - 0s 1ms/step - loss: 1.1883 -
accuracy: 0.7889
22/22 [==============================] - 0s 1ms/step
Epoch 1/20
58/58 [==============================] - 1s 6ms/step - loss: 0.6706 -
accuracy: 0.6814 - val_loss: 0.6495 - val_accuracy: 0.7282
Epoch 2/20
58/58 [==============================] - 0s 4ms/step - loss: 0.5594 -
accuracy: 0.8683 - val_loss: 0.5698 - val_accuracy: 0.7573
Epoch 3/20
58/58 [==============================] - 0s 4ms/step - loss: 0.4020 -
accuracy: 0.8877 - val_loss: 0.5091 - val_accuracy: 0.7670
```

```
Epoch 4/20
58/58 [==============================] - 0s 4ms/step - loss: 0.2851 -
accuracy: 0.9195 - val_loss: 0.4907 - val_accuracy: 0.7524
Epoch 5/20
58/58 [==============================] - 0s 4ms/step - loss: 0.2189 -
accuracy: 0.9303 - val_loss: 0.5191 - val_accuracy: 0.7913
Epoch 6/20
58/58 [==============================] - 0s 4ms/step - loss: 0.1768 -
accuracy: 0.9476 - val_loss: 0.5211 - val_accuracy: 0.7816
Epoch 7/20
58/58 [==============================] - 0s 6ms/step - loss: 0.1486 -
accuracy: 0.9498 - val_loss: 0.5220 - val_accuracy: 0.7816
22/22 [==============================] - 0s 2ms/step - loss: 0.6124 -
accuracy: 0.8006
22/22 [==============================] - 0s 1ms/step
Epoch 1/20
29/29 [==============================] - 1s 12ms/step - loss: 0.6810 -
accuracy: 0.6172 - val_loss: 0.6659 - val_accuracy: 0.7136
Epoch 2/20
29/29 [==============================] - 0s 7ms/step - loss: 0.6148 -
accuracy: 0.8494 - val_loss: 0.6230 - val_accuracy: 0.7427
Epoch 3/20
29/29 [==============================] - 0s 7ms/step - loss: 0.5170 -
accuracy: 0.8834 - val_loss: 0.5633 - val_accuracy: 0.7524
Epoch 4/20
29/29 [==============================] - 0s 8ms/step - loss: 0.4012 -
accuracy: 0.9104 - val_loss: 0.5205 - val_accuracy: 0.7816
Epoch 5/20
29/29 [==============================] - 0s 7ms/step - loss: 0.3103 -
accuracy: 0.9212 - val_loss: 0.4957 - val_accuracy: 0.7718
Epoch 6/20
29/29 [==============================] - 0s 8ms/step - loss: 0.2489 -
accuracy: 0.9325 - val_loss: 0.4852 - val_accuracy: 0.7961
Epoch 7/20
29/29 [==============================] - 0s 7ms/step - loss: 0.2030 -
accuracy: 0.9465 - val_loss: 0.4879 - val_accuracy: 0.7913
Epoch 8/20
29/29 [==============================] - 0s 7ms/step - loss: 0.1756 -
accuracy: 0.9519 - val_loss: 0.4850 - val_accuracy: 0.7913
Epoch 9/20
29/29 [==============================] - 0s 7ms/step - loss: 0.1520 -
accuracy: 0.9590 - val_loss: 0.4924 - val_accuracy: 0.7864
Epoch 10/20
29/29 [==============================] - 0s 7ms/step - loss: 0.1304 -
accuracy: 0.9611 - val_loss: 0.4964 - val_accuracy: 0.7913
Epoch 11/20
29/29 [==============================] - 0s 7ms/step - loss: 0.1200 -
accuracy: 0.9649 - val_loss: 0.5285 - val_accuracy: 0.7913
22/22 [==============================] - 0s 2ms/step - loss: 0.6197 -
```

```
accuracy: 0.7962
22/22 [==============================] - 0s 2ms/step
Epoch 1/20
15/15 [==============================] - 1s 16ms/step - loss: 0.6854 -
accuracy: 0.5751 - val_loss: 0.6745 - val_accuracy: 0.7087
Epoch 2/20
15/15 [==============================] - 0s 8ms/step - loss: 0.6422 -
accuracy: 0.8202 - val_loss: 0.6521 - val_accuracy: 0.7670
Epoch 3/20
15/15 [==============================] - 0s 8ms/step - loss: 0.5888 -
accuracy: 0.8855 - val_loss: 0.6183 - val_accuracy: 0.7621
Epoch 4/20
15/15 [==============================] - 0s 8ms/step - loss: 0.5188 -
accuracy: 0.9028 - val_loss: 0.5821 - val_accuracy: 0.7476
Epoch 5/20
15/15 [==============================] - 0s 8ms/step - loss: 0.4440 -
accuracy: 0.9136 - val_loss: 0.5472 - val_accuracy: 0.7524
Epoch 6/20
15/15 [==============================] - 0s 8ms/step - loss: 0.3709 -
accuracy: 0.9239 - val_loss: 0.5194 - val_accuracy: 0.7767
Epoch 7/20
15/15 [==============================] - 0s 8ms/step - loss: 0.3115 -
accuracy: 0.9276 - val_loss: 0.5017 - val_accuracy: 0.7718
Epoch 8/20
15/15 [==============================] - 0s 8ms/step - loss: 0.2688 -
accuracy: 0.9390 - val_loss: 0.4899 - val_accuracy: 0.7816
Epoch 9/20
15/15 [==============================] - 0s 9ms/step - loss: 0.2306 -
accuracy: 0.9438 - val_loss: 0.4890 - val_accuracy: 0.7864
Epoch 10/20
15/15 [==============================] - 0s 8ms/step - loss: 0.2035 -
accuracy: 0.9487 - val_loss: 0.4831 - val_accuracy: 0.7864
Epoch 11/20
15/15 [==============================] - 0s 8ms/step - loss: 0.1811 -
accuracy: 0.9536 - val_loss: 0.4820 - val_accuracy: 0.7864
Epoch 12/20
15/15 [==============================] - 0s 8ms/step - loss: 0.1644 -
accuracy: 0.9552 - val_loss: 0.4849 - val_accuracy: 0.7816
Epoch 13/20
15/15 [==============================] - 0s 8ms/step - loss: 0.1483 -
accuracy: 0.9600 - val_loss: 0.4911 - val_accuracy: 0.7816
Epoch 14/20
15/15 [==============================] - 0s 8ms/step - loss: 0.1357 -
accuracy: 0.9595 - val_loss: 0.4939 - val_accuracy: 0.7816
22/22 [==============================] - 0s 2ms/step - loss: 0.5584 -
accuracy: 0.8093
22/22 [==============================] - 0s 2ms/step
Epoch 1/20
58/58 [==============================] - 1s 8ms/step - loss: 0.5679 -
```

```
accuracy: 0.7036 - val_loss: 0.5122 - val_accuracy: 0.7767
Epoch 2/20
58/58 [==============================] - 0s 5ms/step - loss: 0.2478 -
accuracy: 0.9028 - val_loss: 0.6173 - val_accuracy: 0.7816
Epoch 3/20
58/58 [==============================] - 0s 5ms/step - loss: 0.1523 -
accuracy: 0.9460 - val_loss: 0.7839 - val_accuracy: 0.8010
Epoch 4/20
58/58 [==============================] - 0s 5ms/step - loss: 0.1052 -
accuracy: 0.9606 - val_loss: 0.8265 - val_accuracy: 0.7961
22/22 [==============================] - 0s 2ms/step - loss: 1.1108 -
accuracy: 0.7744
22/22 [==============================] - 0s 1ms/step
Epoch 1/20
29/29 [==============================] - 1s 11ms/step - loss: 0.5809 -
accuracy: 0.6965 - val_loss: 0.5308 - val_accuracy: 0.7718
Epoch 2/20
29/29 [==============================] - 0s 7ms/step - loss: 0.2497 -
accuracy: 0.9077 - val_loss: 0.5683 - val_accuracy: 0.7864
Epoch 3/20
29/29 [==============================] - 0s 7ms/step - loss: 0.1501 -
accuracy: 0.9438 - val_loss: 0.7410 - val_accuracy: 0.7961
Epoch 4/20
29/29 [==============================] - 0s 6ms/step - loss: 0.0952 -
accuracy: 0.9638 - val_loss: 0.7459 - val_accuracy: 0.8010
22/22 [==============================] - 0s 2ms/step - loss: 0.9448 -
accuracy: 0.7773
22/22 [==============================] - 0s 1ms/step
Epoch 1/20
15/15 [==============================] - 1s 17ms/step - loss: 0.6089 -
accuracy: 0.6857 - val_loss: 0.5468 - val_accuracy: 0.7621
Epoch 2/20
15/15 [==============================] - 0s 9ms/step - loss: 0.2908 -
accuracy: 0.8920 - val_loss: 0.5877 - val_accuracy: 0.7476
Epoch 3/20
15/15 [==============================] - 0s 9ms/step - loss: 0.1541 -
accuracy: 0.9401 - val_loss: 0.6031 - val_accuracy: 0.7913
Epoch 4/20
15/15 [==============================] - 0s 8ms/step - loss: 0.0978 -
accuracy: 0.9708 - val_loss: 0.6391 - val_accuracy: 0.7913
22/22 [==============================] - 0s 2ms/step - loss: 0.7368 -
accuracy: 0.7918
22/22 [==============================] - 0s 1ms/step
Epoch 1/20
58/58 [==============================] - 1s 8ms/step - loss: 0.6668 -
accuracy: 0.6809 - val_loss: 0.6682 - val_accuracy: 0.7524
Epoch 2/20
58/58 [==============================] - 0s 6ms/step - loss: 0.5901 -
accuracy: 0.8348 - val_loss: 1.7546 - val_accuracy: 0.6845
```

```
Epoch 3/20
58/58 [==============================] - 0s 6ms/step - loss: 0.3866 -
accuracy: 0.8726 - val_loss: 1.3649 - val_accuracy: 0.7913
Epoch 4/20
58/58 [==============================] - 0s 6ms/step - loss: 0.2808 -
accuracy: 0.9136 - val_loss: 1.4459 - val_accuracy: 0.7670
22/22 [==============================] - 0s 2ms/step - loss: 1.6295 -
accuracy: 0.7715
22/22 [==============================] - 0s 1ms/step
Epoch 1/20
29/29 [==============================] - 1s 11ms/step - loss: 0.5945 -
accuracy: 0.7030 - val_loss: 0.5576 - val_accuracy: 0.7476
Epoch 2/20
29/29 [==============================] - 0s 7ms/step - loss: 0.3410 -
accuracy: 0.8704 - val_loss: 0.7153 - val_accuracy: 0.7718
Epoch 3/20
29/29 [==============================] - 0s 7ms/step - loss: 0.2967 -
accuracy: 0.9071 - val_loss: 2.1471 - val_accuracy: 0.7524
Epoch 4/20
29/29 [==============================] - 0s 7ms/step - loss: 0.2212 -
accuracy: 0.9271 - val_loss: 1.8267 - val_accuracy: 0.8010
22/22 [==============================] - 0s 2ms/step - loss: 2.8347 -
accuracy: 0.7773
22/22 [==============================] - 0s 2ms/step
Epoch 1/20
15/15 [==============================] - 1s 17ms/step - loss: 0.5837 -
accuracy: 0.7003 - val_loss: 0.6438 - val_accuracy: 0.7573
Epoch 2/20
15/15 [==============================] - 0s 8ms/step - loss: 0.3131 -
accuracy: 0.8737 - val_loss: 0.9662 - val_accuracy: 0.7767
Epoch 3/20
15/15 [==============================] - 0s 8ms/step - loss: 0.1972 -
accuracy: 0.9411 - val_loss: 0.9403 - val_accuracy: 0.7621
Epoch 4/20
15/15 [==============================] - 0s 8ms/step - loss: 0.0989 -
accuracy: 0.9633 - val_loss: 0.9968 - val_accuracy: 0.7961
22/22 [==============================] - 0s 2ms/step - loss: 1.2447 -
accuracy: 0.7817
22/22 [==============================] - 0s 1ms/step
Epoch 1/20
58/58 [==============================] - 1s 13ms/step - loss: 0.6663 -
accuracy: 0.6652 - val_loss: 0.6334 - val_accuracy: 0.7282
Epoch 2/20
58/58 [==============================] - 1s 11ms/step - loss: 0.5144 -
accuracy: 0.8693 - val_loss: 0.5383 - val_accuracy: 0.7573
Epoch 3/20
58/58 [==============================] - 1s 12ms/step - loss: 0.3337 -
accuracy: 0.8985 - val_loss: 0.4977 - val_accuracy: 0.7864
Epoch 4/20
```

```
58/58 [==============================] - 1s 12ms/step - loss: 0.2274 -
accuracy: 0.9287 - val_loss: 0.4921 - val_accuracy: 0.7767
Epoch 5/20
58/58 [==============================] - 1s 13ms/step - loss: 0.1669 -
accuracy: 0.9498 - val_loss: 0.4921 - val_accuracy: 0.7864
Epoch 6/20
58/58 [==============================] - 1s 14ms/step - loss: 0.1319 -
accuracy: 0.9584 - val_loss: 0.5067 - val_accuracy: 0.7864
Epoch 7/20
58/58 [==============================] - 1s 13ms/step - loss: 0.1067 -
accuracy: 0.9676 - val_loss: 0.5295 - val_accuracy: 0.7864
Epoch 8/20
58/58 [==============================] - 1s 10ms/step - loss: 0.0878 -
accuracy: 0.9730 - val_loss: 0.5452 - val_accuracy: 0.7816
22/22 [==============================] - 0s 2ms/step - loss: 0.6475 -
accuracy: 0.7962
22/22 [==============================] - 0s 1ms/step
Epoch 1/20
29/29 [==============================] - 1s 14ms/step - loss: 0.6787 -
accuracy: 0.6253 - val_loss: 0.6618 - val_accuracy: 0.7379
Epoch 2/20
29/29 [==============================] - 0s 11ms/step - loss: 0.5885 -
accuracy: 0.8769 - val_loss: 0.6020 - val_accuracy: 0.7621
Epoch 3/20
29/29 [==============================] - 0s 11ms/step - loss: 0.4549 -
accuracy: 0.9028 - val_loss: 0.5363 - val_accuracy: 0.7767
Epoch 4/20
29/29 [==============================] - 0s 13ms/step - loss: 0.3256 -
accuracy: 0.9174 - val_loss: 0.5017 - val_accuracy: 0.7767
Epoch 5/20
29/29 [==============================] - 0s 14ms/step - loss: 0.2407 -
accuracy: 0.9374 - val_loss: 0.4872 - val_accuracy: 0.7767
Epoch 6/20
29/29 [==============================] - 0s 13ms/step - loss: 0.1886 -
accuracy: 0.9465 - val_loss: 0.4870 - val_accuracy: 0.7961
Epoch 7/20
29/29 [==============================] - 0s 12ms/step - loss: 0.1546 -
accuracy: 0.9536 - val_loss: 0.4907 - val_accuracy: 0.7961
Epoch 8/20
29/29 [==============================] - 0s 13ms/step - loss: 0.1264 -
accuracy: 0.9622 - val_loss: 0.5022 - val_accuracy: 0.7864
Epoch 9/20
29/29 [==============================] - 0s 13ms/step - loss: 0.1105 -
accuracy: 0.9676 - val_loss: 0.5123 - val_accuracy: 0.7864
22/22 [==============================] - 0s 2ms/step - loss: 0.5874 -
accuracy: 0.7933
22/22 [==============================] - 0s 2ms/step
Epoch 1/20
15/15 [==============================] - 1s 32ms/step - loss: 0.6825 -
```

```
accuracy: 0.6085 - val_loss: 0.6721 - val_accuracy: 0.7233
Epoch 2/20
15/15 [==============================] - 0s 13ms/step - loss: 0.6212 -
accuracy: 0.8720 - val_loss: 0.6393 - val_accuracy: 0.7573
Epoch 3/20
15/15 [==============================] - 0s 13ms/step - loss: 0.5453 -
accuracy: 0.9082 - val_loss: 0.5957 - val_accuracy: 0.7816
Epoch 4/20
15/15 [==============================] - 0s 13ms/step - loss: 0.4487 -
accuracy: 0.9239 - val_loss: 0.5534 - val_accuracy: 0.7718
Epoch 5/20
15/15 [==============================] - 0s 15ms/step - loss: 0.3564 -
accuracy: 0.9228 - val_loss: 0.5194 - val_accuracy: 0.7718
Epoch 6/20
15/15 [==============================] - 0s 15ms/step - loss: 0.2848 -
accuracy: 0.9357 - val_loss: 0.5018 - val_accuracy: 0.7816
Epoch 7/20
15/15 [==============================] - 0s 18ms/step - loss: 0.2340 -
accuracy: 0.9433 - val_loss: 0.4990 - val_accuracy: 0.7961
Epoch 8/20
15/15 [==============================] - 0s 15ms/step - loss: 0.1920 -
accuracy: 0.9482 - val_loss: 0.4977 - val_accuracy: 0.8010
Epoch 9/20
15/15 [==============================] - 0s 16ms/step - loss: 0.1664 -
accuracy: 0.9525 - val_loss: 0.4949 - val_accuracy: 0.7864
Epoch 10/20
15/15 [==============================] - 0s 16ms/step - loss: 0.1450 -
accuracy: 0.9568 - val_loss: 0.4966 - val_accuracy: 0.7816
Epoch 11/20
15/15 [==============================] - 0s 21ms/step - loss: 0.1285 -
accuracy: 0.9676 - val_loss: 0.5049 - val_accuracy: 0.7718
Epoch 12/20
15/15 [==============================] - 0s 14ms/step - loss: 0.1161 -
accuracy: 0.9708 - val_loss: 0.5140 - val_accuracy: 0.7816
22/22 [==============================] - 0s 2ms/step - loss: 0.5659 -
accuracy: 0.8035
22/22 [==============================] - 0s 3ms/step
Epoch 1/20
58/58 [==============================] - 1s 14ms/step - loss: 0.5258 -
accuracy: 0.7311 - val_loss: 0.5466 - val_accuracy: 0.7767
Epoch 2/20
58/58 [==============================] - 1s 11ms/step - loss: 0.2241 -
accuracy: 0.9082 - val_loss: 0.6172 - val_accuracy: 0.7913
Epoch 3/20
58/58 [==============================] - 1s 10ms/step - loss: 0.1205 -
accuracy: 0.9514 - val_loss: 0.6576 - val_accuracy: 0.7816
Epoch 4/20
58/58 [==============================] - 1s 11ms/step - loss: 0.0738 -
accuracy: 0.9719 - val_loss: 0.7546 - val_accuracy: 0.7913
22/22 [==============================] - 0s 2ms/step - loss: 0.8795 -
```

```
accuracy: 0.7933
22/22 [==============================] - 0s 2ms/step
Epoch 1/20
29/29 [==============================] - 1s 19ms/step - loss: 0.5497 -
accuracy: 0.7068 - val_loss: 0.5473 - val_accuracy: 0.7573
Epoch 2/20
29/29 [==============================] - 0s 15ms/step - loss: 0.2237 -
accuracy: 0.9201 - val_loss: 0.5935 - val_accuracy: 0.7816
Epoch 3/20
29/29 [==============================] - 0s 15ms/step - loss: 0.1166 -
accuracy: 0.9525 - val_loss: 0.6429 - val_accuracy: 0.8058
Epoch 4/20
29/29 [==============================] - 0s 12ms/step - loss: 0.0724 -
accuracy: 0.9768 - val_loss: 0.7226 - val_accuracy: 0.7718
22/22 [==============================] - 0s 2ms/step - loss: 0.8109 -
accuracy: 0.7846
22/22 [==============================] - 0s 2ms/step
Epoch 1/20
15/15 [==============================] - 1s 20ms/step - loss: 0.5835 -
accuracy: 0.7009 - val_loss: 0.5110 - val_accuracy: 0.7379
Epoch 2/20
15/15 [==============================] - 0s 12ms/step - loss: 0.2402 -
accuracy: 0.9055 - val_loss: 0.5750 - val_accuracy: 0.8155
Epoch 3/20
15/15 [==============================] - 0s 15ms/step - loss: 0.1237 -
accuracy: 0.9530 - val_loss: 0.6285 - val_accuracy: 0.8010
Epoch 4/20
15/15 [==============================] - 0s 24ms/step - loss: 0.0809 -
accuracy: 0.9687 - val_loss: 0.8310 - val_accuracy: 0.8058
22/22 [==============================] - 0s 3ms/step - loss: 1.1050 -
accuracy: 0.7860
22/22 [==============================] - 0s 2ms/step
Epoch 1/20
58/58 [==============================] - 1s 11ms/step - loss: 0.7478 -
accuracy: 0.6847 - val_loss: 0.6646 - val_accuracy: 0.7330
Epoch 2/20
58/58 [==============================] - 1s 9ms/step - loss: 0.6405 -
accuracy: 0.8542 - val_loss: 1.5366 - val_accuracy: 0.7427
Epoch 3/20
58/58 [==============================] - 1s 15ms/step - loss: 0.7177 -
accuracy: 0.8785 - val_loss: 2.4383 - val_accuracy: 0.7573
Epoch 4/20
58/58 [==============================] - 1s 10ms/step - loss: 0.4956 -
accuracy: 0.9087 - val_loss: 1.9228 - val_accuracy: 0.7718
22/22 [==============================] - 0s 2ms/step - loss: 3.6944 -
accuracy: 0.7598
22/22 [==============================] - 0s 2ms/step
Epoch 1/20
29/29 [==============================] - 1s 21ms/step - loss: 0.6991 -
```

```
accuracy: 0.6944 - val_loss: 0.8881 - val_accuracy: 0.7476
Epoch 2/20
29/29 [==============================] - 0s 10ms/step - loss: 0.4490 -
accuracy: 0.8699 - val_loss: 1.2636 - val_accuracy: 0.7573
Epoch 3/20
29/29 [==============================] - 0s 9ms/step - loss: 0.2475 -
accuracy: 0.9201 - val_loss: 1.7929 - val_accuracy: 0.7427
Epoch 4/20
29/29 [==============================] - 0s 17ms/step - loss: 0.1850 -
accuracy: 0.9509 - val_loss: 2.0254 - val_accuracy: 0.7573
22/22 [==============================] - 0s 2ms/step - loss: 2.4580 -
accuracy: 0.7627
22/22 [==============================] - 0s 3ms/step
Epoch 1/20
15/15 [==============================] - 1s 34ms/step - loss: 0.6223 -
accuracy: 0.6911 - val_loss: 0.6470 - val_accuracy: 0.7573
Epoch 2/20
15/15 [==============================] - 0s 19ms/step - loss: 0.3092 -
accuracy: 0.8909 - val_loss: 1.1694 - val_accuracy: 0.7718
Epoch 3/20
15/15 [==============================] - 0s 14ms/step - loss: 0.1697 -
accuracy: 0.9352 - val_loss: 1.1545 - val_accuracy: 0.7913
Epoch 4/20
15/15 [==============================] - 0s 12ms/step - loss: 0.1128 -
accuracy: 0.9611 - val_loss: 1.6043 - val_accuracy: 0.7767
22/22 [==============================] - 0s 2ms/step - loss: 2.0863 -
accuracy: 0.7889
22/22 [==============================] - 0s 2ms/step

Optimal Hyperparameters: {'neurons': 128, 'learningRate': 0.001,
'batchSize': 128}
Highest Accuracy: 80.93%
Classification Report for Best Model:
              precision    recall  f1-score   support

           0       0.79      0.82      0.80       331
           1       0.82      0.80      0.81       356

    accuracy                           0.81       687
   macro avg       0.81      0.81      0.81       687
weighted avg       0.81      0.81      0.81       687

                                             Message
0                The weather outside is not pleasant.
1                                     Team is amazing
2  This novel is truly captivating, a literary ma...
3  The cityscape at night gives life to the sayin...
4  The presentation was monotonous. The guest spe...
5  The city park is unremarkable. The botanical g...
6  The smartphone's battery life is disappointing...
```

```
7  The homework assignment is tedious. The extra ...
8                    The homework assignment is easy
9  The sunrise this morning was absolutely breath...

                                             Message
0                          weather outside pleasant .
1                                        team amazing
2   novel truly captivating , literary masterpiece .
3  cityscape night give life saying , " city slee...
4  presentation monotonous . guest speaker , , de...
5  city park unremarkable . botanical garden near...
6  smartphone battery life disappointing . late m...
7  homework assignment tedious . extra credit pro...
8                            homework assignment easy
9  sunrise morning absolutely breathtaking , pain...

1/1 [==============================] - 0s 48ms/step
y_pred [0, 1, 1, 0, 0, 1, 0, 0, 1, 1]
```