```
Binary classification using DNN: classify movie review based on text context of
reviews.(IMDB dataset)

from tensorflow.keras.datasets import imdb

(x_train, y_train), (x_test, y_test) = imdb.load_data(num_words= 10000)

print("Train shape: ", x_train.shape)
print("Test shape: ", x_test.shape)

print("Train shape: ", y_train.shape)
print("Test shape: ", y_test.shape)

print(x_train[1])

print(y_train[1])

vocab = imdb.get_word_index()
print(vocab['the'])

Class_name = ['Negative', 'Positive']
reverse_index = dict([(value, key) for (key, value) in vocab.items()])

def decode(review):
    text = ""
    for i in review:
        text = text + reverse_index[i]
        text += " "
    return text

decode(x_train[1])

def showlen():
    print("Length of first training sample: ", len(x_train[0]))
    print("Length of first training sample: ", len(x_train[1]))
    print("Length of first training sample: ", len(x_test[0]))
    print("Length of first training sample: ", len(x_test[1]))
showlen()

from tensorflow.keras.preprocessing.sequence import pad_sequences
x_train = pad_sequences(x_train, value = vocab['the'], padding = 'post', maxlen
= 256)
x_test = pad_sequences(x_test, value = vocab['the'], padding = 'post', maxlen =
256)

showlen()

decode(x_train[1])

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Embedding, GlobalAveragePooling1D

model = Sequential()
model.add(Embedding(10000, 16))
```

```
model.add(GlobalAveragePooling1D())
model.add(Dense(16, activation = 'relu'))
model.add(Dense(1, activation = 'sigmoid'))
model.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics =
['accuracy'])
model.summary()

model.fit(x_train, y_train, epochs = 10, batch_size= 128, validation_data=
(x_test, y_test))

x_test[10]

y_test[10]

import numpy as np
predicted_value = model.predict(np.expand_dims(x_test[10],0))
print(predicted_value)
if predicted_value > 0.5 :
    final_value = 1
else:
    final_value = 0
print(final_value)
print(Class_name[final_value])
```