```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from konlpy.tag import Okt
from tensorflow.keras.models import Sequential
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from sklearn.preprocessing import LabelEncoder
from tensorflow.keras.utils import to_categorical
import pickle

df = pd.read_csv('./crawling_data/naver_news_economy_data.csv')
print(df.head())
df.info()

X = df['titles']
Y = df['category']

label_encoder = LabelEncoder()
labeled_y = label_encoder.fit_transform(Y)
label = label_encoder.classes_

with open("./crawling_data/label_encoder.pickle", "wb") as file:
    pickle.dump(label_encoder, file)

onehot_y = to_categorical(labeled_y)

okt = Okt()
for i in range(len(X)):
    X[i] = okt.morphs(X[i], stem=True)

stopwords = pd.read_csv('./stopwords.csv', index_col=0)
for j in range(len(X)):
    words = []
    for i in range(len(X[j])):
        if len(X[j][i]) > 1:
            if X[j][i] not in list(stopwords['stopword']):  # 불용어 제거
                words.append(X[j][i])
    X[j] = ' '.join(words)

token = Tokenizer()
token.fit_on_texts(X)
tokened_x = token.texts_to_sequences(X)
wordsize = len(token.word_index) + 1
# print(tokened_x)
print(wordsize)

with open('./crawling_data/news_token.pickle', 'wb') as f:
    pickle.dump(token, f)

max = 0
for i in range(len(tokened_x)):
    if max < len(tokened_x[i]):
        max = len(tokened_x[i])
print(max)

x_pad = pad_sequences(tokened_x, max)
```

```python
print(x_pad)

X_train, X_test, Y_train, Y_test = train_test_split(
    x_pad, onehot_y, test_size=0.2)
print(X_train.shape, Y_train.shape)
print(X_test.shape, Y_test.shape)

xy = X_train, X_test, Y_train, Y_test
xy = np.array(xy, dtype=object)
np.save('./crawling_data/news_data_max_{}_wordsize_{}'.format(max, wordsize), xy)
```