

```
1 import pandas as pd
2 import numpy as np
3
4 from sklearn.model_selection import train_test_split
5 from sklearn.preprocessing import LabelEncoder
6
7 from konlpy.tag import Okt
8
9 from tensorflow.keras.preprocessing.text import Tokenizer
10 from tensorflow.keras.preprocessing.sequence import pad_sequences
11 from tensorflow.keras.utils import to_categorical
12
13 import pickle
14
15 from tensorflow.keras.models import load_model
16
17 df = pd.read_csv("./crawling_data/naver_test_headline_news_20240130.csv")
18 print(df.head())
19 print(df.info())
20
21 X = df["titles"]
22 Y = df["category"]
23
24 with open("./crawling_data/label_encoder.pickle", "rb") as file:
25     label_encoder = pickle.load(file)
26
27 label = label_encoder.classes_
28
29 print(label)
30
31 okt = Okt()
32
33 for i in range(len(X)):
34     X[i] = okt.morphs(X[i], stem = True)
35
36 stopwords = pd.read_csv("./stopwords.csv")
37
38 for i in range(len(X)):
39     words = []
40     for j in range(len(X[i])):
41         if len(X[i][j]) > 1:
42             if X[i][j] not in list(stopwords["stopword"]):
43                 words.append(X[i][j])
44
45     X[i] = " ".join(words)
46
47 with open("./crawling_data/news_token.pickle", "rb") as file:
48     token = pickle.load(file)
49
50 tokened_x = token.texts_to_sequences(X)
51
52 for i in range(len(tokened_x)):
53     if len(tokened_x[i]) > 23:
54         tokened_x[i] = tokened_x[i][:23]
55
56 print(tokened_x)
57
```

```
58 x_pad = pad_sequences(tokened_x, 23)
59
60 model = load_model("./crawling_data/economy_category_classification_model_0.
5509442687034607.h5")
61
62 preds = model.predict(x_pad)
63
64 predicts = []
65
66 for pred in preds:
67     most = label[np.argmax(pred)]
68     pred[np.argmax(pred)] = 0
69     second = label[np.argmax(pred)]
70     predicts.append([most, second])
71
72 df["predict"] = predicts
73
74 print(df)
75
76
77 df["0X"] = 0
78
79 for i in range(len(df)):
80     if df.loc[i, "category"] in df.loc[i, "predict"]:
81         df.loc[i, "0X"] = "0"
82     else:
83         df.loc[i, "0X"] = "X"
84
85 print(df["0X"].value_counts())
86
87 print(df["0X"].value_counts() / len(df))
88
89
90
```