Chapter_9_Embedding_a_Machine_Learning_Model_into_a_Web_Applic

March 19, 2024

0.1 Serializing fitted Scikit-Learn Estimators

[6]:

```python
from nltk.corpus import stopwords
stop = stopwords.words('english')
```

[7]:

```python
import numpy as np
import re
from nltk.corpus import stopwords
stop = stopwords.words('english')
def tokenizer(text):
    text = re.sub('<[^>]*>', '', text)
    emoticons = re.findall('(?::|;|=)(?:-)?(?:\)|\(|D|P)',text.lower())
    text = re.sub('[\W]+', ' ', text.lower()) \
    + ' '.join(emoticons).replace('-', '')
    tokenized = [w for w in text.split() if w not in stop]
    return tokenized
```

[8]:

```python
def stream_docs(path):
    with open(path, 'r', encoding='utf-8') as csv:
        next(csv) # skip header
        for line in csv:
            text, label = line[:-3], int(line[-2])
            yield text, label
```

[9]:

```python
def get_minibatch(doc_stream, size):
    docs, y = [], []
    try:
        for _ in range(size):
            text, label = next(doc_stream)
            docs.append(text)
            y.append(label)
    except StopIteration:
        return None, None
    return docs, y
```

[10]:

```
from sklearn.feature_extraction.text import HashingVectorizer
from sklearn.linear_model import SGDClassifier
vect =␣
 ↪HashingVectorizer(decode_error='ignore',n_features=2**21,preprocessor=None,tokenizer=tokeni
clf = SGDClassifier(loss='log', random_state=1, max_iter=1)
doc_stream = stream_docs(path='movie_data.csv')
```

[11]:

```
import pyprind
pbar = pyprind.ProgBar(45)
classes = np.array([0, 1])
for _ in range(45):
    X_train, y_train = get_minibatch(doc_stream, size=1000)
    if not X_train:
        break
    X_train = vect.transform(X_train)
    clf.partial_fit(X_train, y_train, classes=classes)
    pbar.update()
```

```
0% [##############################] 100% | ETA: 00:00:00
Total time elapsed: 00:00:28
```

[12]:

```
import pickle
import os
dest = os.path.join('movieclassifier', 'pkl_objects')
```

[13]:

```
if not os.path.exists(dest):
    os.makedirs(dest)
pickle.dump(stop,open(os.path.join(dest, 'stopwords.pkl'),'wb'),protocol=4)
pickle.dump(clf,open(os.path.join(dest, 'classifier.pkl'), 'wb'),protocol=4)
```

Go to the movieclassifier directory and run the code under untitled.ipynb file

0.2 Setting up an SQLite database for data storage

check moviesclassifier director and untitle ipynb file

1 Turning the movie classifier into a web application

[1]:

```
# run app.py in current location
```

[ ]:

```
```