HW4: Fake News Detection

309554001 劉雨恩

1. Data Preprocessing

1. 將 `train.csv` 、 `test.csv` 和 `sample_submission.csv` 的資料 (`id` , `text` , `label`) 提取出來。

```python
import pandas as pd

# read data
fp = open(f'{data_path}{mode}.csv')
data_lines = fp.readlines()
fp.close()

# data preprocess
data_df = pd.DataFrame([], columns=['id', 'text', 'label'])
if mode == 'train':
    for i, l_i in enumerate(data_lines):
        if i == 0:
            continue
        try:
            dict_i = {}
            text_i, label = l_i.split('\t')

            dict_i['id'] = [str(i)]
            dict_i['text'] = [text_i.strip()]
            dict_i['label'] = [int(label.strip())]
            data_df = pd.concat([data_df, \
                        pd.DataFrame.from_dict(dict_i, orient='columns')])
        except:
            pass
else:
    label_data = pd.read_csv(f'{data_path}sample_submission.csv')

    for i, (test_li, label_i) in \
            enumerate(zip(data_lines, label_data['label'])):
        if i == 0:
            continue
        dict_i = {}
        id, text_i = test_li.split('\t')

        dict_i['id'] = [id.strip()]
        dict_i['text'] = [text_i.strip()]
        dict_i['label'] = [int(label_i)]
        data_df = pd.concat([data_df, \
                    pd.DataFrame.from_dict(dict_i, orient='columns')])
```

2. 利用 `spacy` 提供的停頓詞列表來去除停頓詞，並使用 `TfidfVectorizer` 將文字資料型態轉換成向量。

```python
import spacy
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer

spacy.load('en_core_web_sm')
spacy_stopwords = spacy.lang.en.stop_words.STOP_WORDS

def data_preprocess(data_path, mode, stop_words):
    ...
    tv = TfidfVectorizer(stop_words=stop_words, max_features=10000)
    X_tf = tv.fit_transform(data_df['text']).toarray()
```

2. Model Training

訓練三種模型 `XGBClassifier` 、 `GradientBoostingClassifier` 和 `LGBMClassifier` ，並計算 `Accuracy` 、 `Precision` 、 `Recall` 和 `F-measure` 。

```python
from xgboost import XGBClassifier
from sklearn.ensemble import GradientBoostingClassifier
from lightgbm import LGBMClassifier
from sklearn.metrics import accuracy_score, precision_score, \
                            recall_score, f1_score

def fit_model(target_path, model_type, x_train, y_train, \
              x_test, y_test, test_id):
    # model predict
    if model_type == 'XGBoost':
        model = XGBClassifier(n_estimators=100, max_features=100, \
                            &amp;nbsp; max_depth=5, learning_rate=0.1)
    elif model_type == 'GBDT':
        model = GradientBoostingClassifier(n_estimators=100, \
                                           max_features=100, \
                                           max_depth=5, learning_rate=0.1)
    elif model_type == 'LightGBM':
        model = LGBMClassifier(n_estimators=100, num_leaves=100, \
                               max_depth=5, learning_rate=0.1)

    model.fit(x_train, y_train)
    y_pred = model.predict(x_test)

    # evaluate model
    acc = accuracy_score(y_test, y_pred)
    prec = precision_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred)
    f_measure = f1_score(y_test, y_pred)
    eval_str = f'Accuracy: {acc}, Precision: {prec}, \
                Recall: {recall}, F-measure: {f_measure}\n'
    print(eval_str)

    # submission
    submission = pd.DataFrame({
        "id": test_id,
        "label": y_pred
    })
    sub_path = f'{target_path}pred_{model_type}.csv'
    print(sub_path)
    submission.to_csv(sub_path, index=False)

    return eval_str
```

3. Results

	XGBoost	GBDT	LightGBM
Accuracy	0.497	0.502	0.497
Precision	0.4	0.424	0.4
Recall	0.036	0.023	0.036
F-measure	0.066	0.043	0.066