一、本周研究内容

1. 实现了过采样中的 SMOTE 算法，并在 baseline 方法 KNN 和 SVM 上进行了实验。其中，SVM 实验结果与上周实现的基于 bootstrap 的 oversampling 方法比较相近，KNN 在 SMOTE 上表现不好。

2. 在 Pytorch 上复现了 TextCNN 模型。

二、项目实施当前状态

采用 resampling 的方法解决了数据集类别不均衡的问题。下一步继续实验更多的 Model，如 TextCNN，TextRNN 等。

三、本周成果

1. SMOTE 算法

SMOTE 属于 oversampling 算法中的一种，它的基本思想是对少数类样本进行分析并根据少数类样本通过 K 近邻生成新的正例添加到数据集中。相比于基于 bootstrap 的 oversampling 方法，SMOTE 降低了过拟合风险。而相比于 downsampling 方法，其没有丢失反例数据，也不会像 downsampling 一样易于造成很大的偏差。SMOTE 核心算法如下。

```
[86]: text_train, text_test, label_train, label_test = \
      train_test_split(df['full_text'], df['label'], test_size=0.3,random_state=42)

      print(len(text_train), len(text_test), len(text_train) + len(text_test))

      4944 2120 7064

[87]: sm_pipeline = Pipeline([
          ('vect', CountVectorizer()),
          ('tfidf', TfidfTransformer())
      ])

      vectorized_text = sm_pipeline.fit_transform(text_train)

[88]: # smote
      sm = SMOTE(random_state=10)

      text_train_sm, label_train_sm = sm.fit_sample(vectorized_text, label_train)

[90]: print(text_train_sm.shape[0], text_test.shape[0], text_train_sm.shape[0] + text_test.shape[0])

      9272 2120 11392

[109]: label_train_sm.value_counts()

[109]: 1.0    4636
       0.0    4636
       Name: label, dtype: int64
```

图一： SMOTE 过采样算法解决数据分类不平衡

2. 实验结果

模型使用了 baseline 模型 KNN 和 SVM, 文本进行了预处理，并使用了 TFIDF 作为文本向量化方法。

在 SMOTE 处理过的数据集上表现：

	precision	recall	f1-score	support
0.0	0.96	0.04	0.08	1994
1.0	0.06	0.98	0.11	126
accuracy			0.10	2120
macro avg	0.51	0.51	0.10	2120
weighted avg	0.91	0.10	0.08	2120

图二：KNN Classification Report with SMOTE

	precision	recall	f1-score	support
0.0	0.97	0.97	0.97	1994
1.0	0.52	0.49	0.51	126
accuracy			0.94	2120
macro avg	0.74	0.73	0.74	2120
weighted avg	0.94	0.94	0.94	2120

图七：SVM Classification Report with SMOTE

四、下周计划

下周完成 TextCNN 和 TextRNN 的模型，并且用基于 SMOTE 的 Oversampling 的数据集进行实验。