本科实验报告

课程名称：	人工智能
实验名称：	k-NN 分类
专业名称：	保密管理
学生姓名：	武自厚
学生学号：	20336014
实验地点：	东校园实验楼 D502
实验成绩：	
报告时间：	2022 年 5 月 18 日

一、 实验要求

在给定的文本数据集完成文本情感分类训练, 在测试集完成测试, 计算准确率. 需要对上次给的数据集进行重新划分, 训练集: 测试集为 $8:2$.

- 文本的特征可以使用 TF-IDF.
- 利用 k-NN 完成对测试集的分类，并计算准确率.
- 报告中需探究超参 k 对分类准确率的影响.
- 需要提交简要报告以及代码.

二、 实验过程

1. k-NN 原理

k-NN 是应用于机器学习以及数据挖掘领域的一种非常简明的算法. 其内容可以简要概括为:

(1) 输入一定量带有标签的训练集信息.

(2) 输入测试样本.

(3) 在测试样本中按照样本特征的 “距离” 选择离它最近的 k 个训练样本.

(4) 在这 k 个样本中取最多的标签作为测试样本的预测标签.

其中超参 k, 训练集以及特征的选取会影响该算法的效率和准确度.

2. 文本特征 TF-IDF

TF-IDF 即词频-逆文本频率, 它不仅考虑了特定词语的出现频率, 还根据它在整个训练集中出现的次数调整权重, 计算公式如下:

$$\begin{aligned}\mathrm{idf}_{i,j} &= \ln \frac{\#D}{\#\{d_i | d_i \in D, t_{i,j} \in d_i\} + 1} \\ \mathrm{tfidf}_{i,j} &= \mathrm{tf}_{i,j} \cdot \mathrm{idf}_{i,j}\end{aligned}$$

其中 i 为文本编号,j 为单个文本中的词语编号,D 为文本集合.

容易知道, 在不同文本经常出现的词语的权重会减少, 更加着重那些 “独特的词” 的作用.

3. 关键代码

代码中采用 `sklearn` 库中的 `TfidfVectorizer` 来提取语料库. 具体预测代码如下:

```
def maj(x: np.ndarray):
    counts = np.bincount(x)
    majs = np.argwhere(counts == np.max(counts)).flatten()
    # 多个众数时随机选择
```

```
    return np.random.choice(majs)

def dist(x: np.ndarray, y: np.ndarray) -> np.ndarray:
    # Euclid 距离
    return np.sqrt(np.sum(np.power(x - y, 2), axis=1))

def fetch_knn(x_train, test, k):
    dists = dist(x_train, test) # 计算距离
    return dists.argpartition(k)[:k] # 返回距离最小的 k 个

def predict_unit(test, x_train, y_train, k):
    # 获取 knn 的索引
    sample_indices = fetch_knn(x_train, test, k)
    # 通过索引获取标签
    sample_y = y_train[sample_indices]
    # 返回众数
    return maj(sample_y)

def predict(x_train, x_test, y_train, k):
    return np.apply_along_axis(predict_unit, 1, x_test, x_train, y_train, k)
```

三、 实验结果

采用 1000 个句子的训练集以及 246 个句子的测试集来测试代码, 计算出预测成功率. 展示 $k = 0..1000$, 步长为 100 的情况.

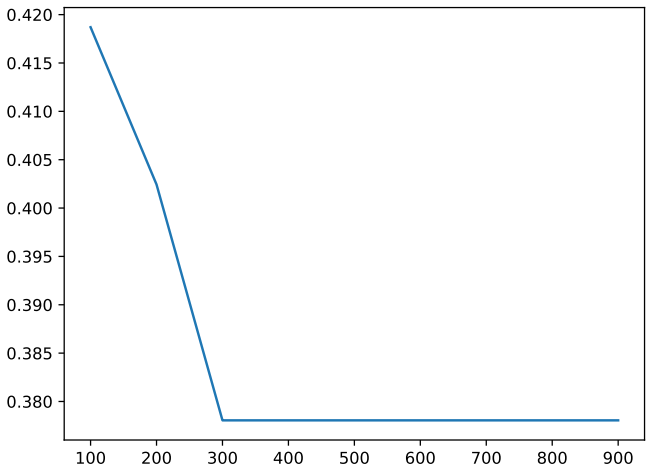

图 1: $k = 0..1000$

可以发现准确率随 k 增加而减少, 选择区间 0..100, 步长为 10 继续测试.

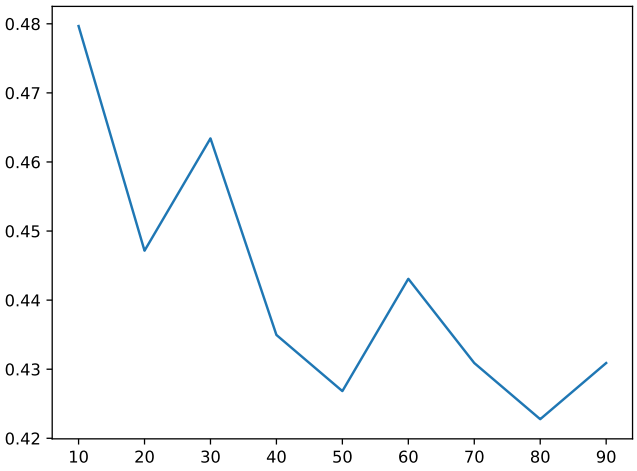

图 2: $k = 0..100$

继续缩小范围, 选择区间 0..40, 步长为 2.

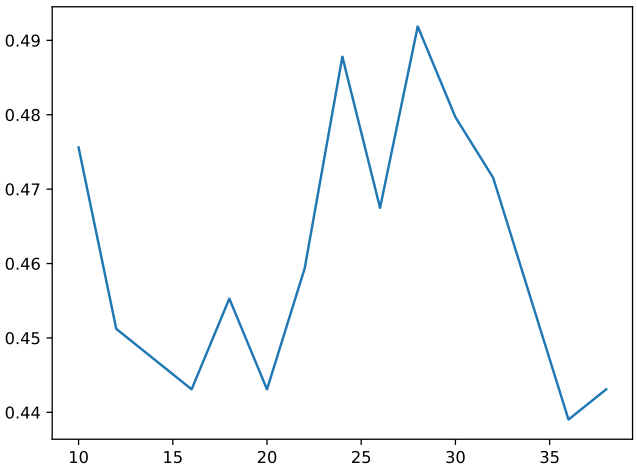

图 3: $k = 0..100$

可以发现算法准确率在 $k = \lfloor\sqrt{N}\rfloor = 31$ 附近达到最大准确率, 符合经验公式.