项目报告：内存故障预测

1. 问题描述

1.1 问题背景

在实际的商业生产中，无法预知的硬件故障，往往会给企业带来很大经济损失，如企业服务器在重大线上活动期间出现故障导致损失以及数据丢失等。因此，尝试收集了一段时间内的内存系统日志、内存故障数据来尝试通过科学的方式来预测某块内存在未来一段时间是否会出现故障，输出预测未来7天会发生内存故障的机器集合，且附带预测时间间隔。

1.2 问题目标

从数据中挖掘出和内存故障相关的特征，训练合适的模型预测内存故障。具体地，在数据中提取出可用的相关特征，首先进行分类，判断系统在未来是否会发生故障，之后在会发生故障的数据当中，进行回归，预测发生故障的时间，

2. 数据获取及预处理

2.1 数据来源

我们的数据来源于阿里云天池挑战赛的数据科学挑战赛：[内存故障预测比赛]

2.2 数据说明

> 数据中含有30个字段，其中字段的含义对应如下：

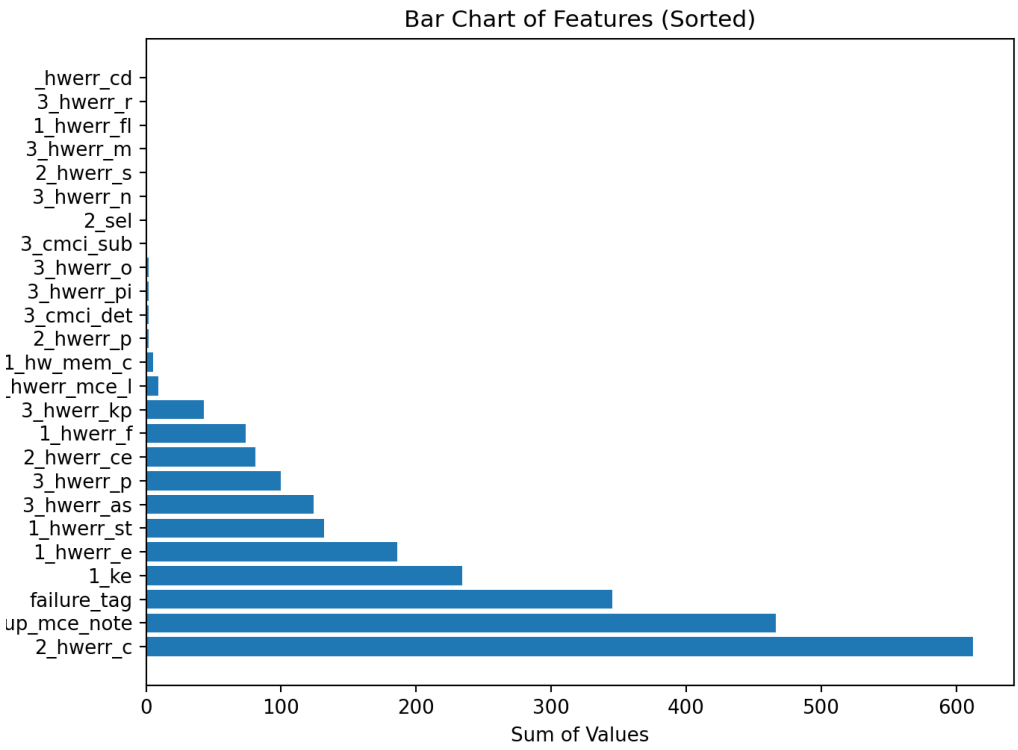
属性	实例	说明
serial_number	server_31576	电脑的编号
collect_time	2019-01-01-00:05:00	日志发生的时间
manufacture	1	制造商
vendor	1.0	供应商
1_hwerr_f	0.0	内核属性

2.3 数据预处理

> 由于数据非常大，超过100万条，因此当我们读取数据的时候，就预先按照时间的接近程度，每五分钟进行一次采样来读取数据，这样读到的数据共有490469条数据，大大减少了噪声的数据，其中没有发生故障的数据有490124条，发生故障的数据有345条，分布非常不平衡。
>
> 在数据中，由于内核的属性的数据，对于没有缺失的部分只有1，因此可以将这些数据当做布尔值考虑，将所有的缺失值都置为0.

3. 数据分析与可视化

> 数据探索性分析的结果，可以使用统计工具，聚类分析等工具
> 使用可视化来展示分析结果



对发生故障的机器的内核做了可视化处理，在预测的时候可以适当增加这些特征的权重

4. 模型选取

> 围绕选题要解决的问题，考虑使用哪些模型来进行挖掘
> 说明选择的理由
>
> 我们使用深度学习和非深度的方法进行训练。
>
> 对于非深度的方法，使用xgboost分类器首先进行分类，判断是否会发生故障，然后使用随机森林的方法进行回归分析，预测发生故障的部分的值是多少
>
> 对于深度的方法，使用不同的网络结构对数据进行判断，此时仍然进行两步的判断，先进行分类，再进行回归。
>
> 我们选择以上方法的原因是这些方法是常见的分类和回归的方法，通常具有较好的表现。

```
# 先进行分类
    clf = XGBClassifier()
    clf.load_model(pretrained_sklearn_weights_path)
    test_predict = clf.predict(X_test.cpu())
    print(test_predict)

    test_ids['predict'] = test_predict
    test_ids = test_ids[test_ids['predict'] == 1]
    test_ids = test_ids.drop('predict', axis=1)

    reg = RandomForestRegressor()
    reg = joblib.load(pretrained_torch_weights_path)

    pre_time = reg.predict(X_test.cpu()[test_predict == 1]).tolist()
    test_ids['pre_time'] = pre_time
    test_ids['pre_time'] = test_ids['pre_time'].astype(int)
```

```
    final_test = test_ids.drop_duplicates(subset=['serial_number'],
keep='first', inplace=False)
```

5. 系统交互设计

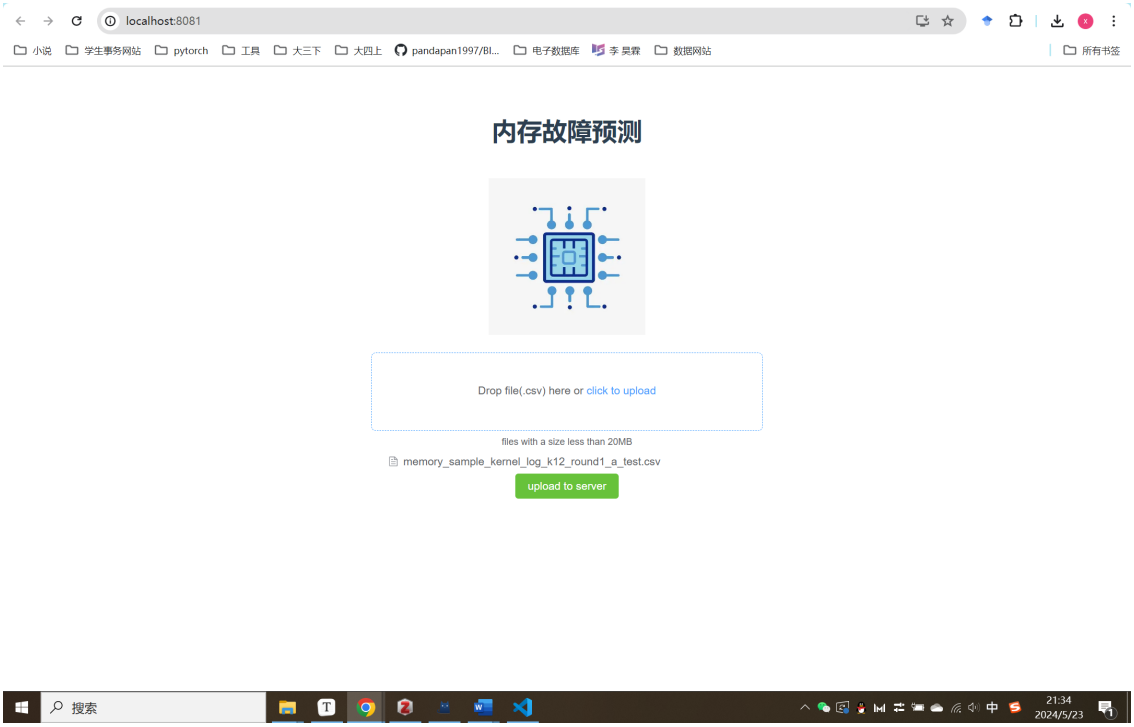
5.1 前端页面总览

前端界面使用Vue3语言，实现了 “click to upload”上传文件的功能，然后使用python的flask框架完成后端部分，后端获取前端输入的文件数据，调用模型接口，获取预先训练好的权重文件，然后返回给前端。返回的值为一个result.csv的文件，包含了电脑的型号、当前的时间、以及会发生故障的时间。

5.2 预测过程展示

1. 点击 “click to upload” 上传测试文件

2. 将测试文件上传到界面



3. 点击 “upload to server” 将测试文件传入后端

5.3 后端返回结果

6. 总结

本次试验完成了对内核故障的预测，首先对数据进行了深入的分析，对数据中的型号等数据进行了剔除，然后对数据进行了可视化的处理。之后首先使用分类器对结果进行分类，获取电脑是否会发生故障。之后使用随机森林对结果进行回归，预测电脑的故障原理。