**流量分类实验报告**

学院：

班级：

学号：

姓名：

# 实验背景

网络流量分类是指，利用某种算法构造一个分类模型，并用该分类模型对收集到的各种应用程序的网络流数据进行分类识别，分类识别的结果是某种应用程序或者应用层协议，又或者是根据QoS要求划分的某种业务类型。影响分类准确率的主要是分类模型，在不同的时代和不同的应用场景下不同分类技术构造的分类模型的分类效果也不一样。

决策树（Decision Tree）是在已知各种情况发生概率的基础上，通过构成决策树来求取净现值的期望值大于等于零的概率，评价项目风险，判断其可行性的决策分析方法，是直观运用概率分析的一种图解法。由于这种决策分支画成图形很像一棵树的枝干，故称决策树。在机器学习中，决策树是一个预测模型，他代表的是对象属性与对象值之间的一种映射关系。Entropy = 系统的凌乱程度，使用算法ID3, C4.5和C5.0生成树算法使用熵。这一度量是基于信息学理论中熵的概念。

决策树是一种树形结构，其中每个内部节点表示一个属性上的测试，每个分支代表一个测试输出，每个叶节点代表一种类别。

分类树（决策树）是一种十分常用的分类方法。它是一种监督学习，所谓监督学习就是给定一堆样本，每个样本都有一组属性和一个类别，这些类别是事先确定的，那么通过学习得到一个分类器，这个分类器能够对新出现的对象给出正确的分类。这样的机器学习就被称之为监督学习。

# 实验内容

## 2.1 使用文档及运行样例

### 2.1.1 项目结构

图形用户界面, 表格

描述已自动生成

### 2.1.2 使用方法

首先运行flow\_combine.py得到流的信息（五元组和报文长度），共三个文件：chat.csv、video.csv和web.csv。

再运行label.py得到流对应的标签，同样为三个文件：chat-label.csv、video-label.csv和web-label.csv。

之后运行net\_streams\_classification.py进行分类并打印准确率。

## 2.2 UML图与主要数据结构

flow\_combine.py：

图形用户界面, 文本

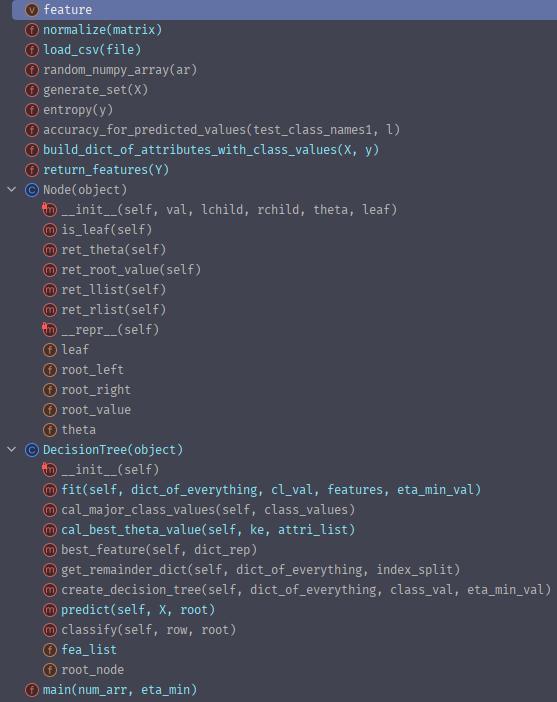
描述已自动生成

label.py:

文本

描述已自动生成

MultiCliassDTree.py:



net\_streams\_classification.py:

文本

描述已自动生成

## 2.3 主要算法说明

### 2.3.1 任务一：处理提供的流量数据集

在flow\_combine.py文件中，extractFlow()函数的功能是提取传入的流列表中的五元组信息和报文长度序列。

在定义的Flow类中的成员变量中已经存储了所需要的信息：src\_ip，dst\_ip，src\_port，dst\_port，trans\_layer\_proto以及pkt\_size\_seq，接下类需要合并以上信息。

在ppt的示例中可以看到共两列，第一列是以列表形式存储的五元组信息，第二列是列表形式的报文长度序列。

以下为代码实现：

# 提取五元组和对应的报文长度序列  
def extractFlow(*flow\_list*):  
 session\_list = []  
  
 for idx in range(len(*flow\_list*)):  
 '''   
 补全此处代码  
 此处需要提取每条流的五元组和对应的报文长度序列  
 session\_list的每一行对应一条流，第一列是五元组组成的列表，第二列是报文长度序列组成的列表  
 '''  
 five\_tup = [*flow\_list*[idx].src\_ip, *flow\_list*[idx].dst\_ip, *flow\_list*[idx].src\_port, *flow\_list*[idx].dst\_port,  
 *flow\_list*[idx].trans\_layer\_proto]  
 session\_list.append([five\_tup, *flow\_list*[idx].pkt\_size\_seq])  
  
 return session\_list

### 2.3.2 任务二：计算信息增益

分为三个步骤，首先根据当前阈值cur\_theta将数据划分进values\_less\_than\_theta\_list和values\_greater\_than\_theta\_list两个列表中，由于data是经过排序的，和原来的index不再对应，所以应当遍历attri\_list与当前阈值比较。

接下类计算每个划分的熵，直接调用前面已经写好的entropy()函数，传入两个存放value的列表即可。

最后计算信息增益，按照README中提供的公式来计算，并对计算结果进行判断，若得到了更大的信息增益，则对信息增益、阈值、索引、类别值。

以下为代码实现：

'''  
补全此部分代码，实现以下逻辑功能：  
 1. 根据当前阈值划分数据  
 2. 计算每个划分的熵  
 3. 计算当前阈值的信息增益, 如果需要，更新最佳阈值  
'''  
# 1. 根据当前阈值划分数据  
for idx, val in enumerate(*attri\_list*):  
 if val[0] < cur\_theta:  
 index\_less\_than\_theta\_list.append(idx)  
 values\_less\_than\_theta\_list.append(val[1])  
 else:  
 index\_greater\_than\_theta\_list.append(idx)  
 values\_greater\_than\_theta\_list.append(val[1])  
# 2. 计算每个划分的熵  
entropy\_less\_than\_theta = entropy(values\_less\_than\_theta\_list)  
entropy\_greater\_than\_theta = entropy(values\_greater\_than\_theta\_list)  
# 3. 计算当前阈值的信息增益, 如果需要，更新最佳阈值  
cur\_info\_gain = entropy\_of\_par\_attr - (  
 entropy\_less\_than\_theta \* (len(index\_less\_than\_theta\_list) / float(len(*attri\_list*)))) - (  
 entropy\_greater\_than\_theta \* (  
 len(index\_greater\_than\_theta\_list) / float(len(*attri\_list*))))  
if cur\_info\_gain > max\_info\_gain:  
 max\_info\_gain = cur\_info\_gain  
 theta = cur\_theta  
 best\_index\_left\_list = index\_less\_than\_theta\_list  
 best\_index\_right\_list = index\_greater\_than\_theta\_list  
 class\_labels\_list\_after\_split = values\_less\_than\_theta\_list + values\_greater\_than\_theta\_list

### 2.3.3 任务三：流量分类

在依次运行完flow\_combine.py和label.py得到共6个csv文件后，运行net\_streams\_classification.py进行流分类，打印准确率。此部分已提供代码，结果详见2.4.3节。

### 2.3.4 任务四(可选)：实验优化

考虑添加一些特征值，使之能够更好的反应出报文长度的分布。在原有特征值的基础上添加了相关系数、自相关系数和互相关系数作为特征：

c\_dict['correlation'] = np.corrcoef(*flow*, *flow*)[0, 1] # 相关系数  
c\_dict['auto\_correlation'] = np.correlate(*flow*, *flow*)[0] # 自相关系数  
c\_dict['cross\_correlation'] = np.correlate(*flow*, *flow*[::-1])[0] # 互相关系数

除此之外还需要修改main()函数中的特征数n\_features为14，get\_characteristics()函数开头部分的c\_dict的定义可以简化为：

c\_dict = {}

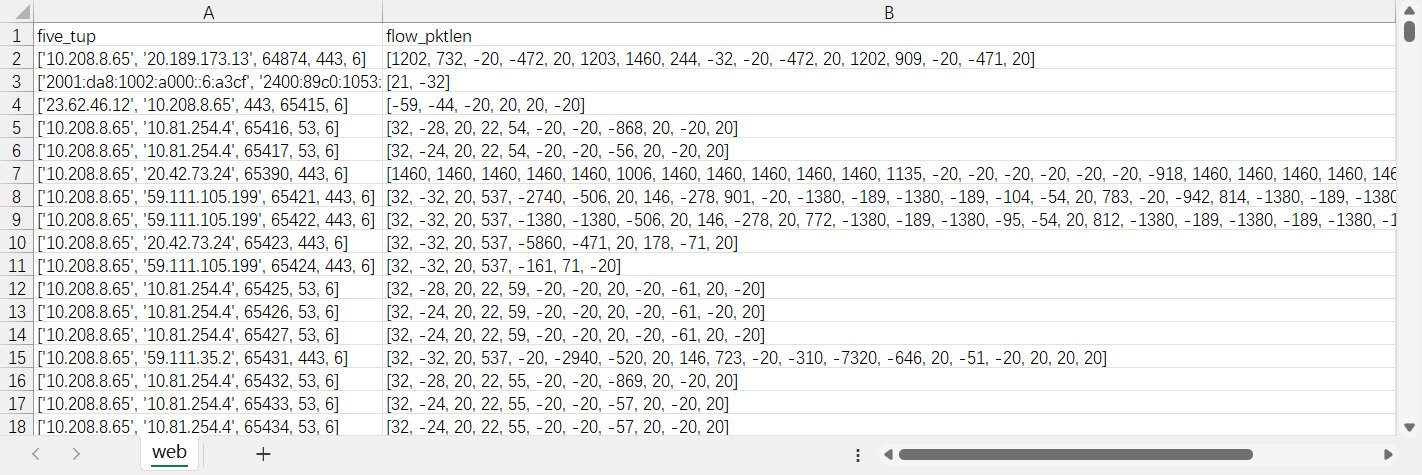
结果详见2.4.4节。

## 2.4 结果展示

### 2.4.1 任务一：处理提供的流量数据集

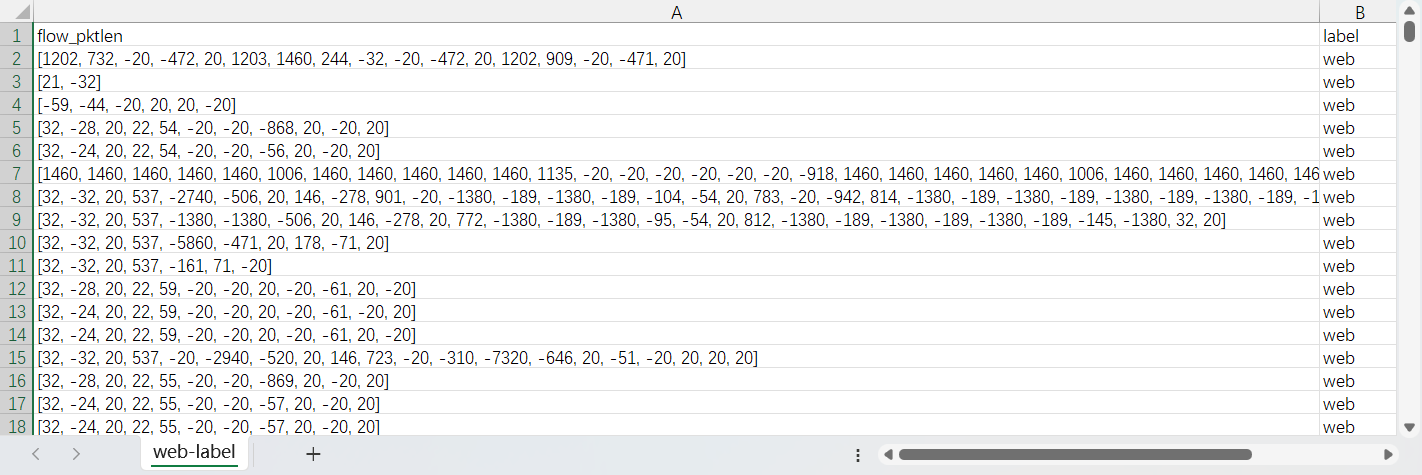
运行flow\_combine.py得到流的信息（五元组和报文长度），共三个文件：chat.csv、video.csv和web.csv。

web.csv：



运行label.py得到流对应的标签，同样为三个文件：chat-label.csv、video-label.csv和web-label.csv。

web-label.csv：



### 2.4.2 任务二：计算信息增益

在控制台打印出每次循环中的阈值、信息增益以及更新情况：

文本

描述已自动生成

图片包含 户外, 键盘, 桌子, 大

描述已自动生成

可以看到信息增益是在不断增大的。

### 2.4.3 任务三：流量分类

打印出每一折的预测值和真实值的对比结果以及准确率，如下：

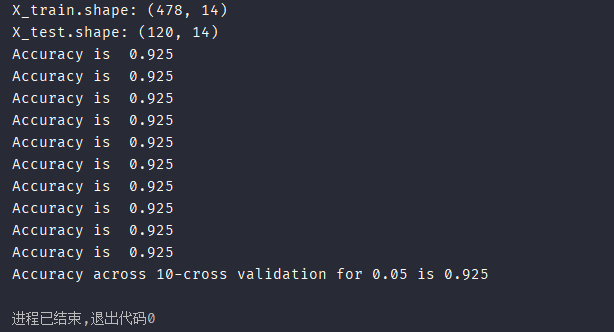
文本

描述已自动生成

在多次测试中，准确率一般在80~91之间波动。

### 2.4.4 任务四(可选)：实验优化

添加了3个特征值后，准确率有所提高，在几次测试中最高可到96.67%：



文本

描述已自动生成

文本

描述已自动生成

# 实验总结

在本次实验中使用到了上一次实验中的组流的方法，重新回顾了一个流中所包含的信息。并且学习了决策树、信息增益、特征提取等知识，在选做部分能够通过自己的多次尝试，观察增加、删除一些特征值后准确率的变化，分析出哪些特征值能够更好的反映出流中报文的分布情况。