## 基于决策树的垃圾邮件分类器的设计与实现

1. **开发环境：**
   1. Windows 10 64 位系统 / macOS High Sierra 10.13.2
   2. Python 3.6.4 (64 位)，需要安装下列依赖库

可以直接 pip install xxx，附上开发时用的版本

* + 1. jieba – 0.39 中文分词
    2. numpy – 1.14.0 数据操作
    3. django – 2.0.1 WebUI 的框架
    4. scipy -1.0.0 和 scikit-learn (sklean) – 0.19.1 现有的决策树算法的库，主要用来对比的
    5. graphviz – 0.8.2 可视化 sklearn 生成的决策树，如果要导出为 pdf，Windows 系统需要安装 graphviz 的运行程序，macOS 则可以通过 brew install graphviz 安装
    6. matplotlib – 2.1.2 可视化自己实现的决策树
  1. PyCharm 2017.3.2，可直接创建 Django 项目，免去许多配置。也可以选择打开一个项目，选择这个工程文件夹即可。

为保证正确运行，请尽量使用相同的开发环境。 涉及到python 的路径设置，请在 Pycharm中调试时，选择正确的运行路径。

注意：涉及到中文文本的处理，请使用 UTF-8 编码！

1. **项目结构：（基于 Django 开发的Web界面）**
2. data 文件夹，经过清洗过的 trec06c 中的数据，因为源数据中有的含有乱码，而且数量巨大，这里只保留了约 6000 条有效数据。
3. dt\_spam，管理整个 Web 的前后端交互、请求、渲染等
4. dt，后端的处理，主要实现了包括数据清洗、数据特征提取、决策树分类器的训练、测试等功能，为前端提供逻辑处理和数据，具体代码在下一部分介绍。
5. SpamDT 文件夹，Django 项目的设置
6. Static 文件夹，网页端使用到的 css 样式和一些 js 代码，用于渲染网页
7. Templates 文件夹，html 网页代码，通过 Django 注入数据，作为前端 UI
8. Index 文件，来自于 trec06c 中，是对数据集中的邮件是否是垃圾邮件的标签

还有一些由代码生成的数据文件，简要说明如下：

* 1. scipy\_tree.pdf，由sklearn 生成的决策树
  2. tree.pickle，自己实现的决策树模型文件，可以直接加载
  3. training.txt，经过特征提取的训练数据集
  4. vocab.txt，根据词频生成的一个中文词的字典

1. **主要代码文件说明**

本项目是 Django 框架开发的，也可以抛开网页 UI 直接运行代码，命令行输出。主要通过 main.py 运行，它是程序的入口。

* 1. 决策树实现相关

dt 文件中的源代码大多数都可以单独测试，即那些下面有 \_\_main\_\_ 函数的

* my\_dt.py，基于最大信息熵、最大信息增益和最大信息增益比三种决策树算法的实现，是整个项目的主要功能
* pre\_process.py，对数据进行预处理，主要完成了特征抽取工作
* prepare.py，对 trec06c 数据集的数据进行清理，将清洗过的数据放在 data 文件夹下。注意：只需要运行一次，在运行时，请将 trec06c 数据集的压缩包也解压在该项目的文件夹中（所给代码中已经删去，但保留了 data 数据）
* scipy\_dt.py，调用第三方的工具包实现的决策树算法，主要是用来观察数据特征，便于自己调整模型的参数等
* view\_dt.py，对自己实现的决策树进行可视化展示
* word2vec.py，原本想使用 word2vec 做特征抽取，但是效果不佳，所以废弃了…
  1. Django 框架，网页 UI
* 主要是 SpamDT/dt\_spam 下的 views.py，里面提供了各种请求的处理和前端渲染
* 根目录下的 manage.py，Django 框架的高级配置（数据库，用户等）相关，这里用不到，是创建 Django 项目时自动包含的

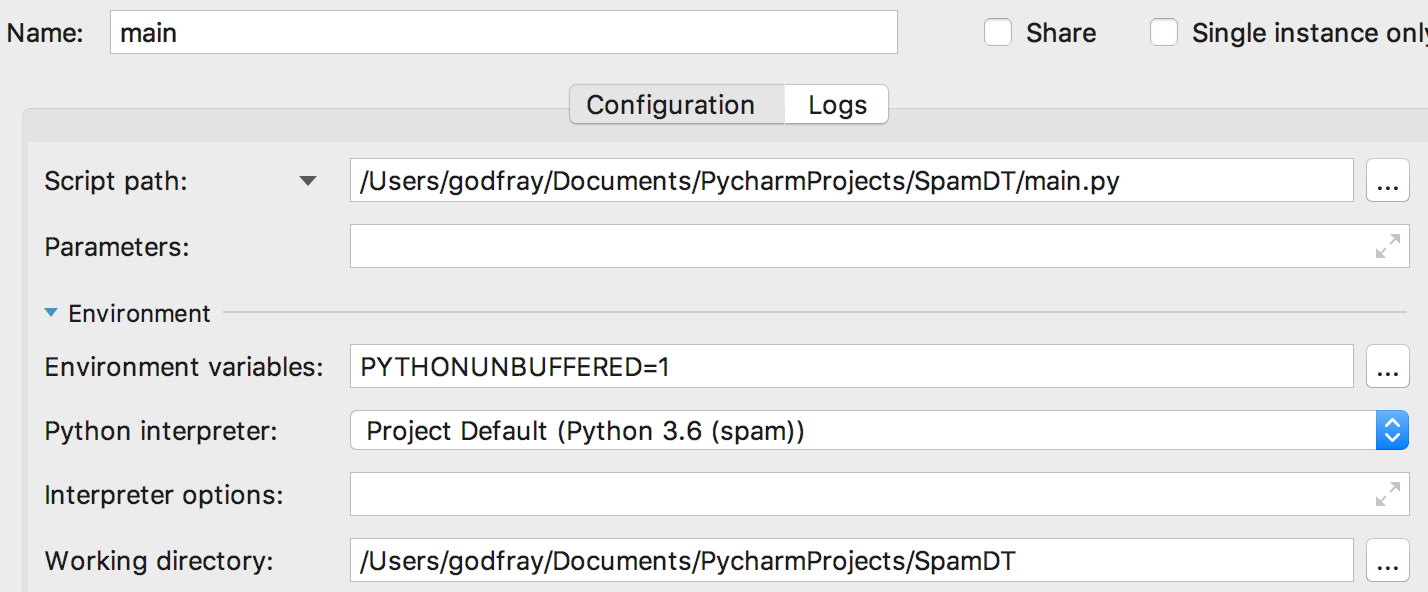
1. **程序运行方式**

可以通过 Pycharm 导入打开，支持命令行运行和 Web UI 运行。为了提高演示的效率，对实际演示的数据集进行了缩减，完整结果请根据注释中 main 函数的地方或者是 views.py 的 index 函数中的对应地方修改。

*注意1：基于命令行的运行方式，需要通过对 main.py 中的部分代码注释，具体操作请根据注释！*

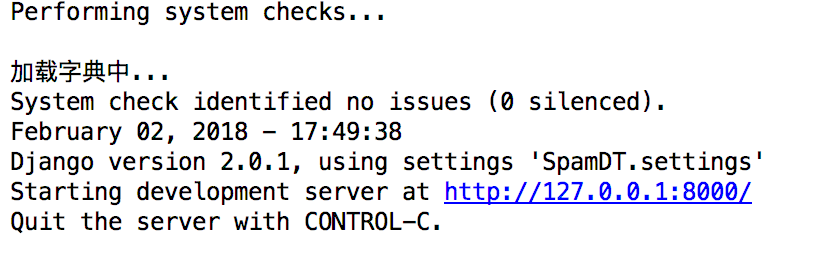
*注意2：由于 python 环境和项目设置的路径不同，需要根据实际情况修改。直接运行可能会出错。有的时候需要重新选择一下 Python interpreter 就没有问题了。*

1. 只使用命令行的分类，在 Pycharm 中运行比较方便，也可以在命令行，切换到根目录中，然后输入 “python main.py”运行，等待结果输出。 Pycharm 中的设置主要是 Script path （请指定好自己的 python 路径）和 Working directory（到根目录为止，即 xxx/xxx/SpamDT）:



* 首先对数据进行清洗，选择合法有效的数据，参考 main.py 的 16-22 行代码
* 特征抽取，main.py 的 31-48 行代码，这个部分比较慢，因为数据量还挺大的
* 使用 sklearn 决策树工具，main.py 的 50-60 行，同时需要确保有 training.txt 和 vocab.txt 这两个文件
* 运行自己实现的决策树，main.py 的 65-103 行，同样需要确保有 training.txt 和 vocab.txt 这两个文件，如何进行参数的设置、算法的选择，请参照具体注释
* 决策树分类器分类测试，main.py 的 105-115 行，需要 vocab.txt 文件和决策树模型 tree.pickle 文件
* 绘制自己生成的决策树，main.py 的 117-125 行，需要有决策树模型 tree.pickle 文件

B) Web UI 演示运行，命令行中，在第一层 SpamDT 文件夹下，使用 “python manage.py runserver 8000” 运行，当有如下的提示后，可以访问 <http://127.0.0.1:8000> ，看到项目的 UI 界面，下面的操作请直接在网页上操作即可。



* 主页界面：进行模型的训练，结果展示
* 实例分析：对用户输入的文本进行是否是垃圾邮件的分类

推荐在 Pycharm 中调试与运行，保持默认配置即可。如果有问题，重新选择一下 python 解释器的正确路径。8000 端口被占用的话，也可以改成其他的。