## 垃圾邮件过滤器的设计与实现

1. 开发环境：
   1. Win 10 64位 / macOS High Sierra 10.13.3
   2. Java 1.8.0\_162 64位
   3. Intellij IDEA 2017.3
   4. Maven （Intellij 已经集成的）

为保证正确运行，请尽量使用相同的开发环境，Java 9 的支持还有些问题。理论上支持 Linux 等其他环境。

1. 项目结构（Maven 构建，可以直接导入）：

导入项目时，在 Intellij IDEA 中直接选择打开，如果是 Eclipse，可以选择 Import maven project。之后会自动下载配置，需要等待较长一段时间，等待下载完成。

1. src 文件夹，项目代码，下面详细解释
2. target 文件夹，Maven 生成的项目 class 文件
3. pox.xml，Maven 的配置文件，自动下载各种 lib
4. spam.iml，Intellij 中，该项目的配置文件
5. bayes\_model.txt 由程序生成的 Bayes 分类器的模型文件
6. model.zip，多层感知器模型文件，可以从中读取已经训练好的模型
7. spam.log 日志记录文件
8. spamdata 是垃圾邮件的数据集
9. 代码文件说明

介绍一下主要的项目代码文件

1. src/main/java/…/offline 中，是关于数据读取、模型训练、模型测试的接口
   1. Bayes.java，实现了朴素贝叶斯分类算法进行垃圾邮件的识别
   2. MLP.java 实现了多层感知器分类器对垃圾邮件的识别
   3. MakeFeature.java 实现了从英文邮件文本转换为特征向量的功能
2. src/main/java/com/example/spam/controller 中，是基于 Spring Boot + Thymeleaf 渲染实现的 Web 项目，主要有根据用户选择的算法训练、加载模型，同时还支持实时的垃圾邮件过滤功能。
   1. ClassifyController.java，根据前端接收到的用户输入的邮件文本，进行处理和分析，结合现有的两种模型进行预测是否为垃圾邮件，将结果返回给前端，浏览器显示结果；
   2. HomeController.java 实现了该 Web 项目的主页的逻辑。通过用户选择的算法、分类器，后台进行本地测试后，将测试结果返回给前端，浏览器中得以显示。
3. SpamApplication.java是整个项目运行的入口，具体使用方式可以看里面的注释，运行 Web 项目或只是本地测试。
4. src/main/resources一些资源文件、项目配置和 Web 项目的静态资源
   1. application.properties 是 Spring Boot 项目的相关配置和一些常量设置
   2. logback-spring.xml 是日志记录配置
   3. static 文件夹中是网页相关的 css 样式，js 脚本和一些图片 img 资源
   4. templates 文件夹中是通过 thymeleaf 渲染的 HTML 网页模板
5. 程序运行方式
   1. 准备工作

在 Intellij IDEA 或者是 (My)Eclipse 中导入该 Maven 项目，会根据 pom.xml 自动下载依赖包，时间可能有些长，如果遇到网络状况不佳，自行百度一下怎么在这些开发环境中更换 maven 源为国内的源，等待加载完毕

* 1. 运行 SpanApplication 本地测试程序

根据注释说明，按步骤运行 Bayes、MLP 分类算法的训练、测试。同时也支持某种算法下的单个样例测试。

* 1. 运行网页程序

只需要执行 SpringApplication.run(SpamApplication.class, args); 当项目构建好后，会有如下类似的提示，实际可能略有偏差，

`com.example.spam.SpamApplication: Started SpamApplication in 5.053 seconds (JVM running for 6.119) `

就可以从浏览器访问 localhost:8080 进行操作