

# 一、目录

## 目录

- 一、目录.....1
- 二、需求规格说明.....3
  - 2.1 引言.....3
    - 2.1.1 编写目的.....3
    - 2.1.2 项目背景.....3
  - 2.2 任务概述.....3
    - 2.2.1 目标.....3
    - 2.2.2 运行环境.....3
    - 2.2.3 条件与限制.....3
  - 2.3 数据描述.....4
    - 2.3.1 动态数据.....4
  - 2.4 功能需求.....4
    - 2.4.1 功能划分.....4
    - 2.4.2 功能描述.....4
  - 2.5 运行需求.....4
    - 2.5.1 用户界面.....4
    - 2.5.2 软件接口.....4
  - 2.6 用例.....5
    - 2.6.1 用例图.....5
    - 2.6.2 用例描述.....5
- 三、系统设计说明.....7
  - 3.1 总体设计.....7
    - 3.1.1 处理流程.....7
    - 3.1.2 总体结构和模块外部设计.....7
    - 3.1.3 功能分配（各项功能与程序结构的关系）.....7
  - 3.2 界面设计.....8
    - 3.2.1 样式设计.....8
    - 3.2.2 导航栏设计.....8
    - 3.2.3 路由设计.....8
  - 3.3 算法设计.....8
    - 3.3.1 TF-IDF 算法.....8
    - 3.3.2 LightGBM 算法.....9
  - 3.4 接口设计.....11
    - 3.4.1 外部接口.....11
    - 3.4.2 内部接口.....12
  - 3.5 程序描述.....12
    - 3.5.1 数据预处理.....12

3.5.2	模型训练.....	13
3.5.3	预测结果.....	14
四、	系统测试说明.....	15
4.1	引言.....	15
4.1.1	编写目的.....	15
4.2	测试计划.....	15
4.2.1	测试配置准备.....	15
4.2.2	测试人员准备.....	15
4.2.3	测试进度表.....	16
4.3	测试概要.....	16
4.3.1	测试用例设计.....	16
4.4	测试内容和执行情况.....	18
4.4.1	功能测试.....	18
4.4.2	性能测试.....	19
4.5	覆盖分析.....	20
4.5.1	测试覆盖率.....	20
4.6	缺陷的统计与分析.....	20
4.6.1	缺陷汇总.....	20
4.6.2	缺陷分析.....	21
4.7	测试总结和建议.....	21
4.7.1	测试总结.....	21
4.7.2	建议.....	21
五、	系统演示说明.....	22
5.1	功能.....	22
5.2	运行环境.....	22
5.3	使用说明.....	22
5.3.1	输入.....	22
5.3.2	输出.....	22
5.3.3	界面介绍.....	22
5.3.3	运行步骤.....	24

## 二、需求规格说明

### 2.1 引言

#### 2.1.1 编写目的

需求分析说明书是为软件的开发者能更好的理解开发的需求，避免由于对问题认识的不清或错误理解而增加更多的开发成本。需求分析是软件系统生存期中定义阶段的最后一个步骤。是作为整个软件开发范围的指南，是软件开发人员开发出正确的符合用户要求的软件的重点。是为明确软件需求、安排项目规划与进度、组织软件开发与测试，撰写本文档该文档将最终交给软件具体开发人员进行具体的开发，其针对的对象是软件开发人员。

#### 2.1.2 项目背景

- a. 软件名称：真假职位信息检测；
- b. 软件介绍：本软件根据提供的职位信息来判断这个职位是否真实存在。
- c. 提出背景：由于目前社会上有存在一定程度的职位信息造假现象，在职位信息上存在一定程度的欺诈行为。为了解决这个问题，辨识出哪些职位信息是假的，存在欺诈行为的，哪些职业信息是真的，是确切可信的，这个项目就是为此而诞生的。因此，该真假职位信息检测系统能够使得有欺诈行为的职业信息被暴露出来，解决了职位信息造假欺诈的问题。

### 2.2 任务概述

#### 2.2.1 目标

通过读取文件中的职位信息，根据训练模型判断所输入的职位信息是否真实存在。

#### 2.2.2 运行环境

- a. 支持 Python3 环境；支持 sklearn、flask、pandas、lightgbm 包。
- b. 浏览器（除了 IE 浏览器）。

#### 2.2.3 条件与限制

支持运行环境即可。

## 2.3 数据描述

### 2.3.1 动态数据

- a. 输入数据: test.csv, train.csv
- b. 输出数据: importance.csv, prediction.csv, 图表

## 2.4 功能需求

### 2.4.1 功能划分

- a. 数据预处理
- b. 模型训练
- c. 结果预测

### 2.4.2 功能描述

- a. 数据预处理: 将 csv 文件内不符合规范或者会影响模型结果的数据进行处理, 使其变为正确的数据。
- b. 模型训练: 用从数据预处理得到的 train 集和 test 集进行模型训练, 用 LightGBM 算法训练模型, 将模型保存为\*.pkl 文件, 将元素数据各个元素的重要性导出为\*.csv 文件。
- c. 结果预测: 用训练好的模型和 test 集对数据进行结果预测, 将预测结果导出为\*.csv 文件。

## 2.5 运行需求

### 2.5.1 用户界面

用户界面标题为“真假职位信息检测”, 提供了一个导航栏, 一级菜单为上传文件和结果展示, 上传文件含有一个“选择文件”的二级菜单; 结果展示含有“importance rank”、“前五个职位检测”、“真假职位个数”的二级菜单, 点击相应的菜单会路由到相应的结果图表进行结果展示。

### 2.5.2 软件接口

接口	路径	函数名	传入数据类型	函数功能
接口 1	http://127.0.0.1:5000/receive	receive	*.csv	接收前端上传的

				test 集文件
接口 2	http://127.0.0.1:5000/g2	g2	json	返回各数据元素重要性给前端
接口 3	http://127.0.0.1:5000/g1	g1	json	返回 test 集的预测结果给前端

## 2.6 用例

### 2.6.1 用例图

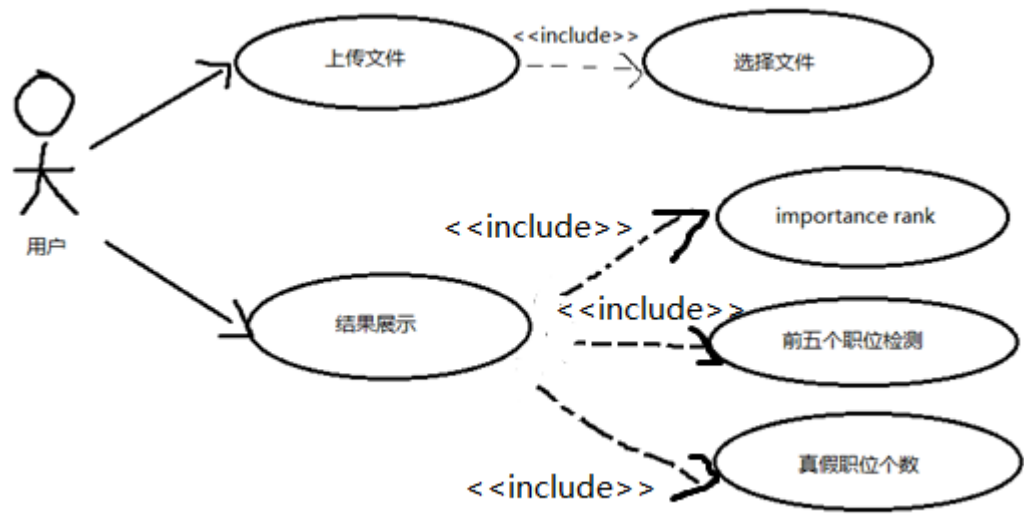


图 1 用例图

### 2.6.2 用例描述

项目	内容描述
用例名称	上传文件
参与者	用户
触发条件	用户点击导航栏中的上传文件
前置条件	系统正常运行
后置条件	文件成功上传
正常流程	1. 用户点击导航栏中的上传文件 2. 用户点击选择文件按钮 3. 用户点击上传文件按钮 4. 本页展示简要结果向用户表明上传成功
扩展流程	用户在按“上传文件”按钮之前，随时都可以重新“选择文件”，之前选择的文件不会对结果造成影响。

项目	内容描述
用例名称	选择文件
参与者	用户
触发条件	用户在上传文件页面点击选择文件按钮
前置条件	系统正常运行
后置条件	系统找到所选择的文件
正常流程	<ol style="list-style-type: none"> <li>1. 用户点击选择文件按钮。</li> <li>2. 系统显示文件名向用户表明系统找到了所选择的文件。</li> </ol>
扩展流程	用户在按“上传文件”按钮之前，随时都可以重新“选择文件”，之前选择的文件不会对结果造成影响。

项目	内容描述
用例名称	结果展示
参与者	用户
触发条件	用户点击导航栏中的结果展示
前置条件	系统正常运行且文件上传成功
后置条件	系统展示结果
正常流程	用户点击导航栏中的结果展示。
扩展流程	用户随时都可以重新选择其他想要查看的结果项目，也可以随时重新上传文件。

项目	内容描述
用例名称	Importance rank
参与者	用户
触发条件	用户点击导航栏结果展示中的“importance rank”菜单
前置条件	系统正常运行且文件上传成功
后置条件	系统展示结果
正常流程	<ol style="list-style-type: none"> <li>1. 用户点击导航栏结果展示中的“importance rank”菜单。</li> <li>2. 系统显示图表向用户展示结果。</li> </ol>
扩展流程	用户随时都可以重新选择其他想要查看的结果项目，也可以随时重新上传文件。

项目	内容描述
用例名称	前五个职位检测
参与者	用户
触发条件	用户点击导航栏结果展示中的“前五个职位检测”菜单
前置条件	系统正常运行且文件上传成功
后置条件	系统展示结果
正常流程	<ol style="list-style-type: none"> <li>1. 用户点击导航栏结果展示中的“前五个职位检测”菜单。</li> <li>2. 系统显示图表向用户展示结果。</li> </ol>
扩展流程	用户随时都可以重新选择其他想要查看的结果项目，也可以随时重新上传文件。

项目	内容描述
用例名称	真假职位个数
参与者	用户
触发条件	用户点击导航栏结果展示中的“真假职位个数”菜单
前置条件	系统正常运行且文件上传成功
后置条件	系统展示结果
正常流程	1. 用户点击导航栏结果展示中的“真假职位个数”菜单。 2. 系统显示图表向用户展示结果。
扩展流程	用户随时都可以重新选择其他想要查看的结果项目，也可以随时重新上传文件。

## 三、系统设计说明

### 3.1 总体设计

#### 3.1.1 处理流程

首先从前端接收用户上传的文件，通过接口传送到后端，后端通过接口获取用户在前端输入的文件，经过数据预处理、模型训练、结果预测得到结果，保存为文件返回给前端，然后前端再获取后端返回的信息处理成表格展示给用户。

#### 3.1.2 总体结构和模块外部设计

##### 3.1.2.1 前端

用户界面设计，通过接口与后端进行交互，将用户输入的信息（\*.csv 文件）传送给后端进行处理，获取后端返回的信息（预测结果\*.csv 文件和元素的重要性\*.csv 文件）处理成表格展示给用户。

##### 3.1.2.2 后端

通过接口获取前端用户输入的信息（\*.csv 文件），经过数据预处理、模型训练、结果预测得到结果（预测结果\*.csv 文件和元素的重要性\*.csv 文件）返回给前端。

#### 3.1.3 功能分配（各项功能与程序结构的关系）

- a. 数据预处理：将 csv 文件内不符合规范或者会影响模型结果的数据进行

处理，使其变为正确的数据。

b. 模型训练：用从数据预处理得到的 `train` 集和 `test` 集进行模型训练，用 `LightGBM` 算法训练模型，将模型保存为 `*.pkl` 文件，将元素数据各个元素的重要性导出为 `*.csv` 文件。

c. 结果预测：用训练好的模型和 `test` 集对数据进行结果预测，将预测结果导出为 `*.csv` 文件。

## 3.2 界面设计

### 3.2.1 样式设计

标题为“真假职位信息检测”，运用 `vue` 框架和 `element-ui` 库、`echarts` 库，构建了一个 `el-container` 容器，并在 `el-header` 上构建了一个横向导航栏，导航栏的一级菜单为：“上传文件”、“结果展示”；“上传文件”一级菜单的二级菜单为“选择文件”二级菜单；“结果展示”为可视化模块，二级菜单为“importance rank”、“前五个职位检测”、“真假职位个数”。点击相应的导航栏会路由到相应的结果图表。

### 3.2.2 导航栏设计

在导航栏内实现了多级菜单，一级菜单为“上传文件”和“结果展示”，二级菜单为“选择文件”、“importance rank”、“前五个职位检测”、“真假职位个数”。一级菜单还嵌入了相应的图标，更具提醒功能。而导航栏在 `App.vue` 里被导入，使得所有页面都会显示导航栏。

### 3.2.3 路由设计

在 `./router/index.js` 注册各个页面的路由，点击相应按钮通过 `@click` 事件调用相应函数，并路由至对应的 `vue` 文件。

## 3.3 算法设计

### 3.3.1 TF-IDF 算法

TF-IDF(Term Frequency - Inverse Document Frequency)是一种用于信息检索、文本挖掘的算法。TF-IDF 算法作为一种统计学的检索算法，可以用于评估一个字或一个词在一份文本或者语料集中的重要性。这个算法由两部分组成，分别是词频和逆向文件概率，一个词在一篇文章的重要性与出现频率成正相关关系，但也会随着它在语料库中出现的频率成负相关关系。其核心思想是：若一个词语在一份文本中出现的频率很高，但在其他文章或者语料集中出现的频率较低，则该词



语拥有较好的代表性，适合作为主题词。

词频(Term Frequency): 表示词语在某一文本中的出现频率，如式(1)所示，其中 $TF_{i,j}$ 代表词语  $i$  在文件  $d_j$  中的词频， $n$  代表词语， $n_{i,j}$  是该词在文件  $d_j$  中出现的次数，分母则是文件  $d_j$  中所有词汇出现的次数总和。

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (1)$$

逆向文件频率(Inverse Document Frequency): 某一词语的逆向文件频率，可以通过总文件数目除以包含该词语文件的数目，再把商的结果取对数即可得到，如式(2)所示，其中 $idf_i$ 代表词语  $i$  的 IDF， $|D|$  是语料库中的总文件数目， $|\{j: t_i \in d_j\}|$  是含有该词语的文件数目。当该词语不存在于语料集中时，会导致分母为 0，从数学理论上来说是不可以进行对数运算的，故一般情况下使用 $|\{j: t_i \in d_j\}| + 1$ 。

$$idf_i = \log \frac{|D|}{|\{j: t_i \in d_j\}| + 1} \quad (2)$$

所以一个词语用 TF-IDF 算法表示为：

$$TF - IDF = TF \times IDF \quad (3)$$

### 3.3.2 LightGBM 算法

#### 3.3.2.1 LightGBM 算法简介

Light GBM 是微软亚洲研究院 DMTK 团队在 Github 上开源的性能超越其他 Boosting decision tree 的工具，一经推出就广受好评，广泛运用于各大数据挖掘比赛。与 XGBoost 相比，Light GBM 主要有直方图算法、基于梯度的单边采样算法、互斥特征捆绑算法三方面的优化。

#### 3.3.2.2 基于直方图的决策树算法

直方图算法的基本思想是先把连续的浮点特征值离散化成  $k$  个整数，也就是分桶 bins 思想，例如 $[0,0.1)$ 离散化为 0， $[0.1,0.3)$ 离散化为 1。用离散化后的值代替原来的浮点值进行计算，统计每个 bin 里面的样本个数，根据离散值来寻找最优的切分点。需要切分的个数就等于 bins 的个数减 1，而 XG-Boost 的切分点个数等于样本取值的不同个数减 1，需要计算的是样本的特征值、样本的一阶导和二阶导。后续计算 bins 里样本一阶导之和与二阶导之和时，计算量也会相应小很多。Light GBM 只会保存离散化后的值，在内存占用上也会小很多。

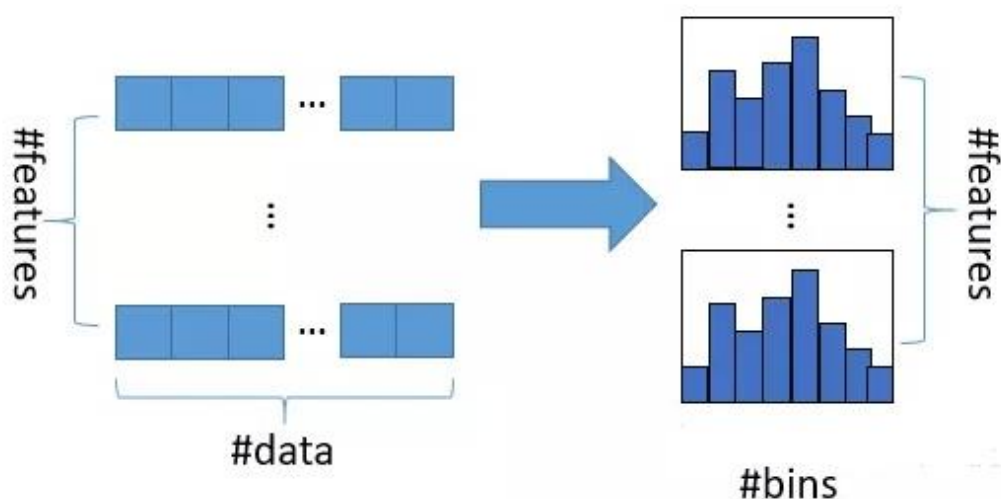


图 2 直方图算法

### 3.3.2.3 直方图做差加速

Light GBM 的另一个优点是使用直方图进行差分加速。当节点分裂成两个时，右边子节点的直方图就可以直接通过上方父亲节点的直方图减去左边其兄弟节点的直方图得到，用这种做差的方法可以使其在运算速度上加倍。在构建完一个叶子节点的直方图后，Light GBM 可以先计算直方图较小的叶节点，然后利用直方图做差来得到直方图大的叶节点，这样，可以以相对较小的成本得到它兄弟叶的直方图。例如，将某个特征的取值分成 4 个桶，每个桶里面的样本个数分别为 4、4、5、3，左边子节点直方图每个桶中的样本个数分别为 1、1、2、1，那么右边子节点的直方图样本个数分别为 3、3、3、2。



图 3 直方图做差加速

### 3.3.2.4 带深度限制的 Leaf-wise 算法

在直方图算法之上，Light GBM 还进行了进一步的优化。按层生长(level-wise)的决策树生长策略是大多数 GBDT 工具使用的策略，而 Light GBM 不再使用该策略，转而使用带有深度限制的按叶生长(leaf-wise)的算法。按层生长(level-wise)同一层的所有节点都要做分裂，分裂完成后再根据需要进行剪枝，因为它毫无区别地对待同一层中的叶子，需要分裂计算的节点过多，所以 Level-wise 效率相对较低，它带来了许多不必要的开销。同一层的其他很多叶子节点也根本无需进行分裂。

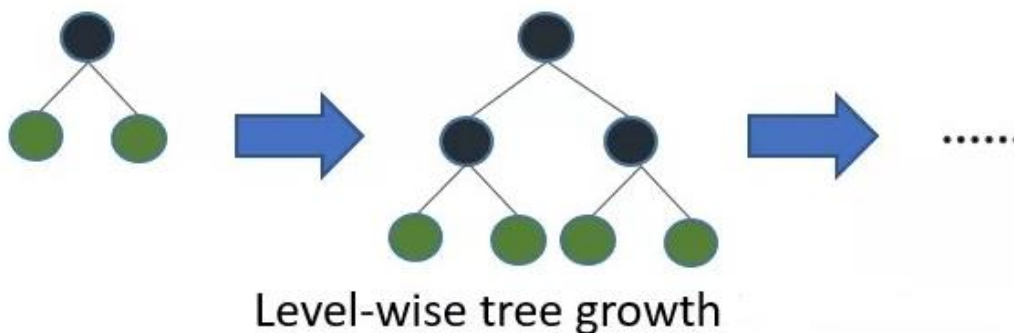


图 4 按层生长的决策树

Leaf-wise 以降低模型损失最大化为目的，是一种更高效的策略。与按层生长的决策树相比，按叶生长的决策树只需要在分裂增益收益最大的节点处进行分裂，其他的节点处不需要进行分裂，可以得到更小的误差，但同时分裂的次数会增加，生长出的决策树过于庞大，容易产生过拟合。为了解决这个弊端，Light GBM 可以设置一个决策树的最大深度。

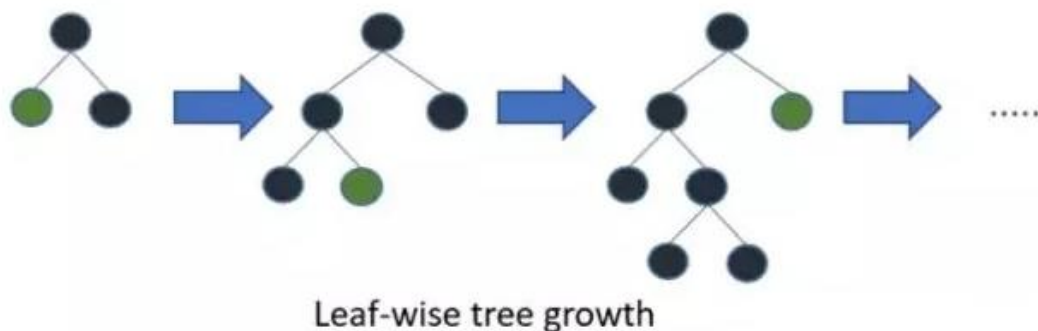


图 5 按叶生长的决策树

## 3.4 接口设计

### 3.4.1 外部接口

#### 3.4.1.1 后端与前端的接口

接口	路径	函数名	传入数据类型	函数功能
接口 1	http://127.0.0.1:5000/receive	receive	*.csv	接收前端上传的 test 集文件
接口 2	http://127.0.0.1:5000/g2	g2	json	返回各数据元素重要性给前端
接口 3	http://127.0.0.1:5000/g1	g1	json	返回 test 集的预测结果给前端

## 3.4.2 内部接口

### 3.4.2.1 后端内部接口

接口	函数名	传入参数类型	返回参数类型	函数功能
接口 1	predict	DataFrame		传入 test 集，加载训练完成的模型对其进行预测，并保存预测结果。
接口 2	process	str	int	传入字符串，返回字符串长度。
接口 3	process1	str	int	传入字符串，返回包含数值。
接口 4	process2	str	int	传入字符串，返回包含数值。
接口 5	get_test_tfidf	DataFrame, str, int	DataFrame	传入 test 集、列名、最大特征数，返回拓展后的 test 集。

## 3.5 程序描述

### 3.5.1 数据预处理

#### 3.5.1.1 功能

将所接收到的\*.csv 文件中会影响模型训练结果准确性的数据进行处理，如把字符串类型转换为小写、处理缺失的数据、对具有明显分类的数据添加标签、通过 TF-IDF 算法处理文本数据。

#### 3.5.1.2 输入项目

\*.csv 文件（train 文件和 test 文件）。

#### 3.5.1.3 输出项目

DataFrame（处理好的 train 集和 test 集）。

### 3.5.1.4 算法

TF-IDF 算法。

TF-IDF 的主要思想是：如果某个单词在一篇文章中出现的频率 TF 高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类。

### 3.5.1.5 程序逻辑

首先从前端接收 train 和 test 的.csv 文件，对文件进行数据预处理：先把字符串类型的标题用 lower() 函数将字符串转换为小写字母，然后处理缺失的数据：对 nan 数据打上一个标记，删去缺失的数据，获取有用的数据（如 0-13000 以为分界，获取 range 的起始与结束），通过 LabelEncoder 对具有明显分类的数据添加标签，应用到 fraudulent 列中（fraudulent 结果若为真，我们则认为这个职位信息具有欺诈成分），最后通过 TF-IDF 算法处理文本数据，得到关键词。然后删去不需要的列，查看行列长度，检查表格，结束数据预处理。

### 3.5.1.6 接口

接口	函数名	传入参数类型	返回参数类型	函数功能
接口 1	process	str	int	传入字符串，返回字符串长度。
接口 2	process1	str	int	传入字符串，返回包含数值。
接口 3	process2	str	int	传入字符串，返回包含数值。

## 3.5.2 模型训练

### 3.5.2.1 功能

用从数据预处理得到的 train 集和 test 集进行模型训练，用 LightGBM 算法训练模型，将模型保存为\*.pkl 文件，将元素数据各个元素的重要性导出为\*.csv 文件。

### 3.5.2.2 输入项目

DataFrame（train 集和 test 集）。

### 3.5.2.3 输出项目

- \*.pkl 模型文件（lgb\_model.pkl）;
- \*.csv 文件（importance.csv）。

### 3.5.2.4 算法

LightGBM 算法。

LightGBM（Light Gradient Boosting Machine）是一个实现 GBDT 算法的框架，支持高效率的并行训练，并且具有更快的训练速度、更低的内存消耗、更好的准确率、支持分布式可以快速处理海量数据等优点。主要思想是利用弱分类器（决策树）迭代训练以得到最优模型。

### 3.5.2.5 程序逻辑

首先从前端接收 train 和 test 的.csv 文件，先对文件进行数据预处理，然后用处理好的数据进行模型训练，然后将模型文件导出为 lgb\_model.pkl、将元素的重要性导出为 importance.csv。

## 3.5.3 预测结果

### 3.5.3.1 功能

用训练好的模型和 test 集对数据进行结果预测，将预测结果导出为\*.csv 文件。

### 3.5.3.2 输入项目

DataFrame（test 集）。

### 3.5.3.3 输出项目

- \*.csv 文件（prediction.csv）

### 3.5.3.4 程序逻辑

调用训练好的模型得到的结果，转换导出为 csv 文件。

3.5.3.5 接口

接口	函数名	传入参数类型	返回参数类型	函数功能
接口 1	predict	DataFrame		传入 test 集，加载训练完成的模型对其进行预测，并保存预测结果。
接口 2	get_test_tfidf	DataFrame, str, int	DataFrame	传入 test 集、列名、最大特征数，返回拓展后的 test 集。

四、系统测试说明

4.1 引言

4.1.1 编写目的

本测试报告为真假职位信息检测项目的测试报告，目的在于总结测试阶段的测试以及分析测试结果，描述系统是否符合需求（或达到功能目标）。预期参考人员包括用户、测试人员、开发人员、项目管理者、其他质量管理人员和需要阅读本报告的高层经理

4.2 测试计划

4.2.1 测试配置准备

测试系统配置	
CPU	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 2.00 GHz
内存	8.00 GB
磁盘空间	250GB
操作系统	Windows 10
浏览器	IE、FireFox、Google Chrome

4.2.2 测试人员准备

测试负责人：制定测试计划，控制测试进度，详细了解被测软件，编写测试用例，执行测试并记录结果，汇报测试结果。

### 4.2.3 测试进度表

项目测试进度表

测试阶段	任务	开始时间	结束时间	加班	增加资源	人员分配
测试环境搭建	搭建测试环境	2022-1-3	2022-1-3	无	无	负责人
编写测试用例	根据软件实际和说明书编写测试用例	2022-1-3	2022-1-3	无	无	负责人
功能测试	测试功能是否满足要求	2022-1-4	2022-1-4	无	无	负责人
性能测试	测试性能是否满足要求	2022-1-4	2022-1-4	无	无	负责人
可靠性、易用性、兼容性测试	测试可靠性、易用性、兼容性是否满足要求	2022-1-4	2022-1-4	无	无	负责人
覆盖率测试	测试覆盖率是否达到正常水平	2022-1-4	2022-1-4	无	无	负责人
编写测试报告	根据计划编写测试报告	2022-1-5	2022-1-5	无	无	负责人

## 4.3 测试概要

### 4.3.1 测试用例设计

#### 4.3.1.1 功能测试

测试编号	JOB_TEST_01					
需求说明	功能测试					
测试环境	测试系统配置					
No	测试目的	测试步骤	预期结果	参考指标	测试结果	备注
1	选择符合要求的文件，上传	选择文件，点击“上传文件”按钮	显示上传成功的文件名	正常	正常	



		钮				
2	查看, 下载上传成功的文件	点击文件名进行下载	显示已下载至本地	正常	正常	
3	上传文件支持中文名称	修改文件名为中文名称, 选择文件, 点击“上传文件”按钮	显示上传成功的文件名	正常	正常	
4	返回各数据元素重要性的 json 数据并展示为图表	文件上传后, 点击“结果展示”按钮, 再点击 importance rank 按钮	显示各数据元素重要性的图表	正常	正常	
5	返回 test 集预测结果的 json 数据并展示为图表	文件上传后, 点击“结果展示”按钮, 再点击“真假职位个数”按钮	显示 test 集预测结果的图表	正常	正常	

#### 4.3.1.2 性能测试

测试编号	JOB_TEST_02					
需求说明	性能测试					
测试环境	测试系统配置					
No	测试目的	测试步骤	预期结果	参考指标	测试结果	备注
1	网速很慢、断网的环境下能正常上传	将网络关闭, 选择文件, 点击“上传文件”按钮	显示上传成功的文件名	正常	正常	
2	预处理、预测过程中 CPU 使用率	文件上传后打开任务管理器, 查看 CPU 使用率	CPU 使用率在正常范围内	0%-2%	0%-0.6%	
3	网速很慢、断网的环境下能返回数据并显示为图表	文件上传后将网络关闭, 点击“结果展示”按钮, 再点击 importance rank 按钮或“真假职位个数”按钮	显示各数据元素重要性的图表或 test 集预测结果的图表	正常	正常	

## 4.4 测试内容和执行情况

### 4.4.1 功能测试

#### 4.4.1.1 功能测试情况概要

模块名称	用例数	用例通过数	问题数	用例通过率
	个	个	个	%
上传文件				
读取文件				
返回数据				

#### 4.4.1.2 上传文件

功能	基本要求	测试情况	测试通过	
			是	否
选择上传	选择符合要求的文件，上传。 上传成功的文件名称显示。	功能实现。	<input checked="" type="checkbox"/>	<input type="checkbox"/>
对上传成功的文件进行操作	查看，下载上传成功的文件。	功能实现。	<input checked="" type="checkbox"/>	<input type="checkbox"/>
文件名及文件路径	上传文件支持中文名称。	功能实现。	<input checked="" type="checkbox"/>	<input type="checkbox"/>

#### 4.4.1.3 读取文件

功能	基本要求	测试情况	测试通过	
			是	否
后端读取文件	成功接收前端文件并读取	功能实现。	<input checked="" type="checkbox"/>	<input type="checkbox"/>

#### 4.4.1.4 返回数据

功能	基本要求	测试情况	测试通过	
			是	否
返回数据	返回各数据元素重要性的 json 数据并展示为图表。 返回 test 集预测结果的 json 数据并展示为	功能实现。	<input checked="" type="checkbox"/>	<input type="checkbox"/>

	图表。			
--	-----	--	--	--

## 4.4.2 性能测试

### 4.4.2.1 性能测试情况概要

项目版本	用例数	用例通过数	问题数	用例通过率
	个	个	个	%
1.0				

### 4.4.2.2 性能测试

序号	用例名称	基本要求	测试情况	测试通过	
				是	否
1	上传	在网络很慢、断网的环境下上传	正常	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	预处理	CPU 使用率在正常范围	正常	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	预测	CPU 使用率在正常范围	正常	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	返回数据	在网络很慢、断网的环境下返回数据	正常	<input checked="" type="checkbox"/>	<input type="checkbox"/>

### 4.4.2.3 可靠性

测试内容	基本要求	测试情况	测试通过	
			是	否
断电	断电或强行关机后重启电脑，不丢失数据。	实现要求。	<input checked="" type="checkbox"/>	<input type="checkbox"/>

### 4.4.2.4 易用性

测试内容	基本要求	测试情况	测试通过	
			是	否
易安装性	安装操作方便。	实现要求。	<input checked="" type="checkbox"/>	<input type="checkbox"/>
易操作性	只有少量按钮，操作方便。	实现要求。	<input checked="" type="checkbox"/>	<input type="checkbox"/>

### 4.4.2.5 兼容性

测试内容	基本要求	测试情况	测试通过	
			是	否

浏览器	在各版本 IE 浏览器中均正常运行	实现要求	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	在 FireFox 中正常运行	实现要求	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	在 Google Chrome 中正常运行	实现要求	<input checked="" type="checkbox"/>	<input type="checkbox"/>
分辨率	在 1920×1080 分辨率下正常运行	实现要求	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	在 1280×960 分辨率下正常运行	实现要求	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	在 800×600 分辨率下正常运行	实现要求	<input checked="" type="checkbox"/>	<input type="checkbox"/>
操作系统	在 Windows 10 操作系统下正常运行	实现要求	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	在 Windows XP 操作系统下正常运行	实现要求	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	在 Windows 7 操作系统下正常运行	实现要求	<input checked="" type="checkbox"/>	<input type="checkbox"/>

## 4.5 覆盖分析

### 4.5.1 测试覆盖率

模块名称	用例个数	执行数	各模块测试覆盖率
代码覆盖率 (使用 pytest-cov)	1	1	71% <pre>C:\Users\lhj\PycharmProjects\distinguishJob&gt;python -m pytest ./main.py ===== test session starts ===== platform win32 -- Python 3.8.5, pytest-6.2.5, py-1.9.0, pluggy-0.13.1 rootdir: C:\Users\lhj\PycharmProjects\distinguishJob plugins: cov-3.0.0 collected 0 items  ----- warnings summary ----- D:\Anaconda3\lib\site-packages\pyreadline\py3k_compat.py:8   D:\Anaconda3\lib\site-packages\pyreadline\py3k_compat.py:8: DeprecationWarning: 'collections.abc' is deprecated, 'collections' instead of from 'collections.abc' is deprecated since Python 3.7, and slated for removal in Python 3.10     return isinstance(x, collections.Callable)  -- Docs: https://docs.pytest.org/en/stable/warnings.html  ----- coverage: platform win32, python 3.8.5-final-0 ----- Name      Stmts   Miss  Cover   Missing ----- main.py      7      2     71%   7, 11 TOTAL        7      2     71%  ===== 1 warning in 1.12s =====</pre>

## 4.6 缺陷的统计与分析

### 4.6.1 缺陷汇总

缺陷编号	模块名称	简要描述	严重程度	附件说明
1	上传文件模块	上传木马文件无法识别	重	
2	上传文件模块	上传文件未限制文件大小	轻	
3	上传文件模块	上传文件未限制文件格式	轻	
4	返回数据模块	图表显示不全	轻	

### 4.6.2 缺陷分析

重要缺陷分析表

缺陷编号	模块名称	简要描述	分析结果	测试通过 (Pass/Fail)
1	上传文件模块	上传木马文件无法识别	系统无法判断文件具体类型	Fail
2	上传文件模块	上传文件未限制文件大小	系统缺乏对文件的限制	Fail
3	上传文件模块	上传文件未限制文件格式	系统缺乏对文件的限制	Fail
4	返回数据模块	图表显示不全	横轴名词太多，图表无法全部显示，只能采取省略的方式	Pass

系统 2 个模块中存在共计 4 个重要缺陷，其中返回数据模块的图表显示不全缺陷采取省略横轴名词的方式通过，存在共计 3 个测试未通过的缺陷。

### 4.7 测试总结和建议

#### 4.7.1 测试总结

“真假职位信息检测系统”在用户现场环境进行功能、性能、可靠性、易用性、兼容性、覆盖率七个方面进行了较为全面、严格、规范的测试。测试结果表明：“真假职位信息检测系统”达到业务需求文档中的要求，并具有以下特点：

1. 系统架构简单。系统结构清晰明确，满足要求。
2. 功能简单。该软件由前端上传文件、后端处理并输出数据等模块组成，输出结果简洁。
3. 系统可靠性高。断电或强行关机后重启电脑、网络异常中断不会损失数据。
4. 系统兼容性好。兼容不同浏览器、分辨率、操作系统。
5. 系统易用性强。软件操作简单，便于使用。
6. 系统代码覆盖率处于正常水平。
7. 系统 2 个模块中存在共计 4 个重要缺陷，其中返回数据模块的图表显示不全缺陷采取省略横轴名词的方式通过，存在共计 3 个测试未通过的缺陷。
8. 测试结论：通过。

#### 4.7.2 建议

1. 系统功能较为简单，可以完善。

2. 增加上传文件时对文件类型、大小、格式的判断，提高系统安全性。

## 五、系统演示说明

### 5.1 功能

检测所输入的职位信息的真假，是否具有欺诈行为。

### 5.2 运行环境

- a. 支持 Python3 环境；支持 sklearn、flask、pandas、lightgbm 包。
- b. 浏览器（除了 IE 浏览器）。

### 5.3 使用说明

#### 5.3.1 输入

##### 5.3.1.1 数据格式

\*.csv 文件，文件内按表头格式填入职位信息。

#### 5.3.2 输出

##### 5.3.2.1 数据格式

echart 表格。

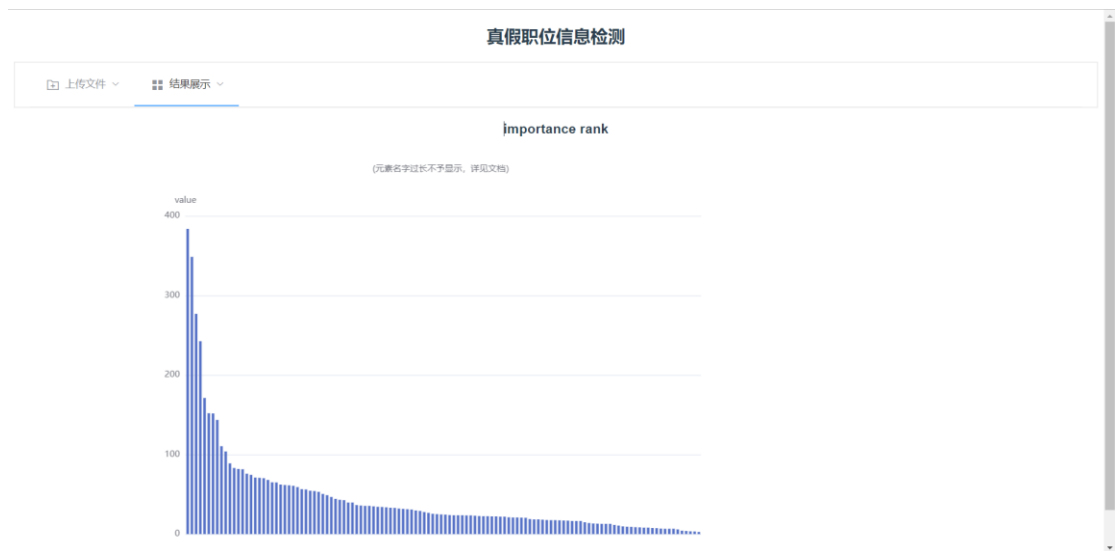
#### 5.3.3 界面介绍

- a. 选择文件页：点击选择文件按钮选择文件，然后点击上传文件按钮上传。



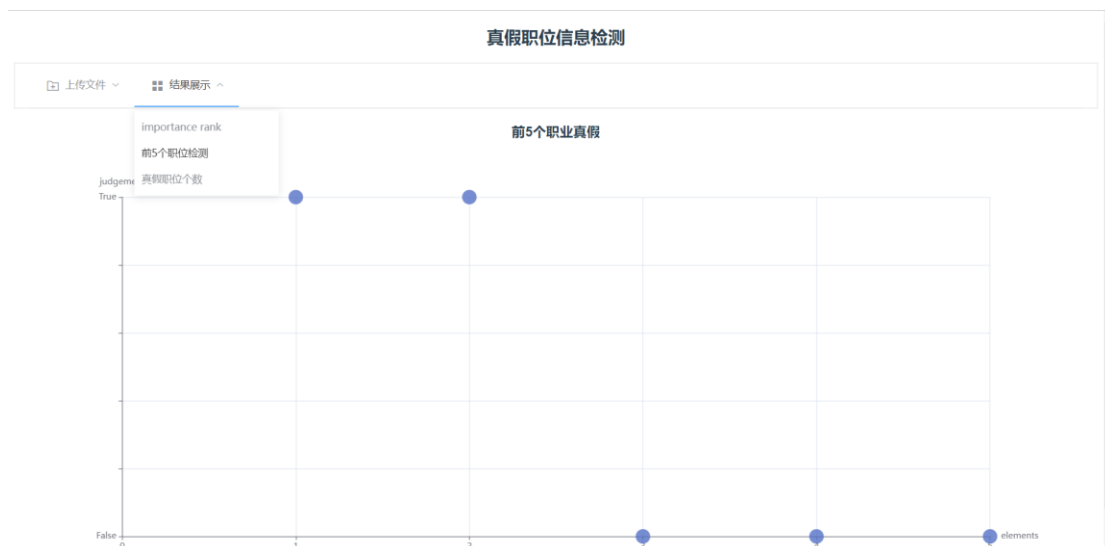
（图 6：选择文件界面）

b. 结果展示：importancerank 页



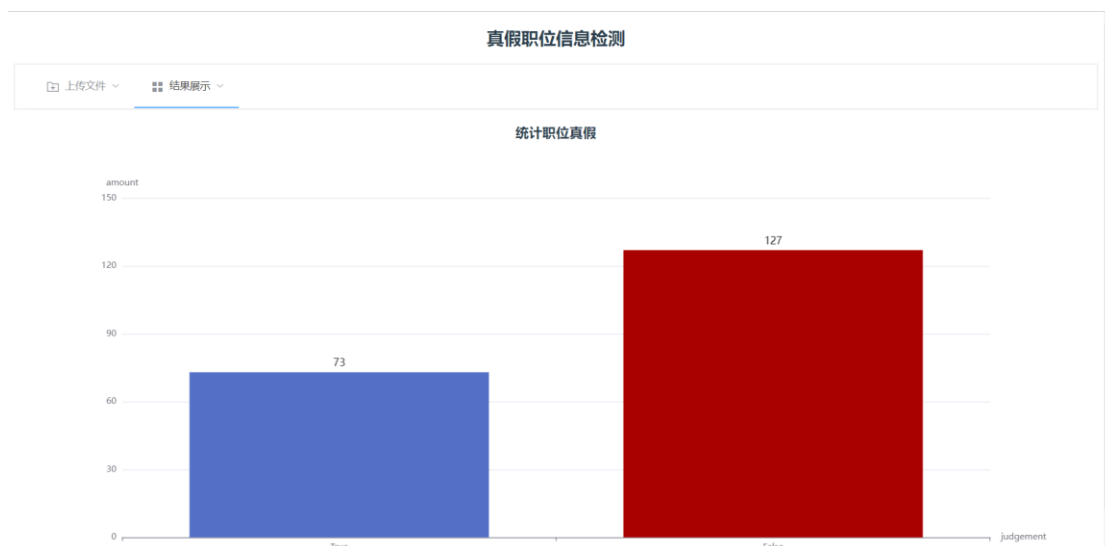
（图 7：重要性结果展示界面）

c. 结果展示：前 5 个职位真假页



(图 8: 前 5 个职位真假展示界面)

d. 结果展示: 真假职位个数页



(图 9: 真假职位个数结果展示界面)

### 5.3.3 运行步骤

- 运行 `distinguishJob/main.py`;
- 打开 `distinguishJob/templates/test.html` 文件;
- 在网页中点击上传文件、选择文件、选择测试集并点击上传文件;
- 点击结果展示的 `importance rank` 菜单查看重要性;
- 点击结果展示的前 5 个职位检测菜单查看前五个职位的检测结果;
- 点击结果展示的真假职位个数菜单查看预测结果。