|  |  |  |
| --- | --- | --- |
| **文件状态**  **[ ] 草稿**  **[ ] 讨论稿**  **[√] 正式发布** | **文件编号** | **010** |
| **文档类型** | **Word** |
| **版 本** | **V1.0** |
| **作 者** | **何钺** |
| **完成日期** | **2019年9月7日** |

项目名称：基于Kaggle数据集的数据分析

文档名称：系统设计文档

**目 录**

[1 引言 1](#_Toc18788365)

[1.1 编写目的 1](#_Toc18788366)

[1.2 术语定义 1](#_Toc18788367)

[1.3 参考资料 1](#_Toc18788368)

[2 任务概述 3](#_Toc18788369)

[2.1 任务目标 3](#_Toc18788370)

[2.2 运行环境 3](#_Toc18788371)

[2.2.1 前端 3](#_Toc18788372)

[2.2.2 后端 4](#_Toc18788373)

[2.3 条件与限制 5](#_Toc18788374)

[2.4 需求规定 5](#_Toc18788375)

[2.4.1 数据预处理 5](#_Toc18788376)

[2.4.2 特征工程 6](#_Toc18788377)

[2.4.3 模型搭建与训练 6](#_Toc18788378)

[2.4.4 数据导入模块 7](#_Toc18788379)

[2.4.5 模型导入模块 7](#_Toc18788380)

[2.4.6 搜索模块 7](#_Toc18788381)

[3 总体设计 7](#_Toc18788382)

[3.1 类图 7](#_Toc18788383)

[3.1.1 类图综述 8](#_Toc18788384)

[3.1.2 类描述 8](#_Toc18788385)

[3.2 用例图 10](#_Toc18788386)

[3.2.1 用例图 10](#_Toc18788387)

[3.2.2 参与者 10](#_Toc18788388)

[3.2.3 用例图描述 11](#_Toc18788389)

[4 接口设计 13](#_Toc18788390)

[4.1 外部接口设计 13](#_Toc18788391)

[4.2 内部接口设计 13](#_Toc18788392)

[5 模型开发过程设计 15](#_Toc18788393)

[5.1 数据预处理 15](#_Toc18788394)

[5.1.1 功能 15](#_Toc18788395)

[5.1.2 输入 15](#_Toc18788396)

[5.1.3 输出 15](#_Toc18788397)

[5.1.4 具体实现 16](#_Toc18788398)

[5.2 特征工程 16](#_Toc18788399)

[5.2.1 功能 16](#_Toc18788400)

[5.2.2 输入 16](#_Toc18788401)

[5.2.3 输出 17](#_Toc18788402)

[5.2.4 具体实现 17](#_Toc18788403)

[5.3 模型搭建 3](#_Toc18788404)

[5.3.1 功能 3](#_Toc18788405)

[5.3.2 输入 3](#_Toc18788406)

[5.3.3 输出 3](#_Toc18788407)

[5.3.4 具体实现 4](#_Toc18788408)

[6 展示系统设计 6](#_Toc18788409)

[6.1 数据导入模块 6](#_Toc18788410)

[6.1.1 功能 6](#_Toc18788411)

[6.1.2 输入 6](#_Toc18788412)

[**6.2** **匹配结果模块** 6](#_Toc18788413)

[6.2.1 功能 6](#_Toc18788414)

[6.2.2 输入 6](#_Toc18788415)

[6.2.3 输出 7](#_Toc18788416)

[**6.3** **评分详情模块** 7](#_Toc18788417)

[6.3.1 功能 7](#_Toc18788424)

[6.3.2 输入 7](#_Toc18788425)

[6.3.3 输出 8](#_Toc18788426)

[6.3.4 程序逻辑 8](#_Toc18788427)

[7 提示信息设计 9](#_Toc18788428)

# 引言

## 编写目的

本文档为“paddle\_paddle”项目设计说明书，是在系统需求分析的基础上，对“paddle\_paddle”项目系统进行设计，主要内容包括系统总体结构、前端设计、系统底层模型设计以及前后端数据传输设计。本文档供程序开发人员阅读，可以为系统开发、软件测试提供依据和参考。

## 术语定义

|  |  |
| --- | --- |
| **术语** | **解释** |
| 用户 | 系统的使用人员，也是该项目的主要服务对象。 |
| 前端 | 指系统中负责视觉设计、UI以及其代码实现的部分，面向开发者。 |
| 后台/后端 | 指系统中负责数据处理、计算及其代码实现的部分，面向开发者。 |

## 参考资料

|  |  |  |
| --- | --- | --- |
| **序号** | **编号** | **文档名称** |
| 1 | paddle\_paddle-001 | 《paddle\_paddle-软件需求规格说明书》 |
| 2 | ISBN 978-7-302-28317-1 | 《软件工程基础》 胡思康编著  清华大学出版社 |

# 任务概述

## 任务目标

“paddle\_paddle”系统的总目标是：根据Kaggle比赛Crowdflower Search Results Relevance的数据集训练模型，开发一款借用此模型来计算和衡量资料库中各搜索结果与用户输入的关键字的相关度的系统，以协助开发搜索算法的企业评估搜索算法的性能。

## 运行环境

### 前端

* **wxPython：**

wxPython是Python语言的一套优秀的GUI图形库，允许Python程序员很方便的创建完整的、功能健全的GUI用户界面。 wxPython是作为优秀的跨平台GUI库wxWidgets的Python封装和Python模块的方式提供给用户的。

就如同Python和wxWidgets一样，wxPython也是一款开源软件，并且具有非常优秀的跨平台能力，能够支持运行在32/64位windows、绝大多数的Unix或类Unix系统、Macintosh OS X下。

wxPython是Python编程语言的一个GUI工具箱。他使得Python程序员能够轻松的创建具有健壮、功能强大的图形用户界面的程序。它是Python语言对流行的wxWidgets跨平台GUI工具库的绑定。而wxWidgets是用C++语言写成的。

wxPython是跨平台的。这意味着同一个程序可以不经修改地在多种平台上运行。现今支持的平台有：32/64位微软Windows操作系统、大多数Unix或类Unix系统、苹果Mac OS X。

由于使用Python作为编程语言，wxPython编写简单、易于理解。

### 后端

* Python：Python是一种计算机程序设计语言。是一种面向对象的动态类型语言，最初被设计用于编写自动化脚本(shell)，随着版本的不断更新和语言新功能的添加，越来越多被用于独立的、大型项目的开发。
* sklearn：scikit-learn(简记sklearn)，是用python实现的机器学习算法库。sklearn可以实现数据预处理、分类、回归、降维、模型选择等常用的机器学习算法。sklearn是基于NumPy, SciPy, matplotlib的。
* TensorFlow：TensorFlow™是一个基于数据流编程（dataflow programming）的符号数学系统，被广泛应用于各类机器学习（machine learning）算法的编程实现，其前身是谷歌的神经网络算法库DistBelief。

Tensorflow拥有多层级结构，可部署于各类服务器、PC终端和网页并支持GPU和TPU高性能数值计算，被广泛应用于谷歌内部的产品开发和各领域的科学研究。

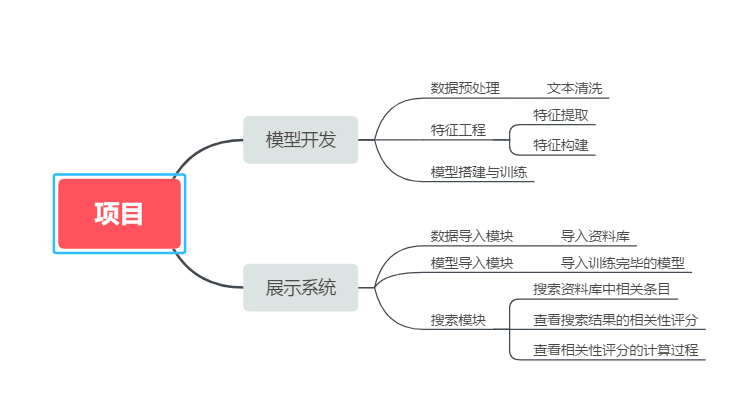
TensorFlow由谷歌人工智能团队谷歌大脑（Google Brain）开发和维护，拥有包括TensorFlow Hub、TensorFlow Lite、TensorFlow Research Cloud在内的多个项目以及各类应用程序接口（Application Programming Interface, API）。自2015年11月9日起，TensorFlow依据阿帕奇授权协议（Apache 2.0 open source license）开放源代码。

## 条件与限制

* Python版本大于等于3.6.8 小于3.7
* TensorFlow版本为1.10

## 需求规定

本系统实现如下功能：



### 数据预处理

训练阶段前对训练数据进行预处理，数据预处理一般包括：

* **删除HTML标签**：由于数据集是从网页中获取，会有很多干扰的无用html标签存在，因此采用bs4库来清除它们
* **同义词替换：**数据中有很多单词其实是同样的意思，因此我们把这些同义词都替换成统一的一个，如：将‘refrigerator’‘fridge’‘freezer’全部替换成‘fridge’
* **其他替换：**数据中有一些组合词是分开的，不便于训练和处理，比如‘16 gb’‘32 g’，因此我们将他们全部改成合并的，如‘16gb’‘32g’。
* **词干提取：**英文中有很多变形，比如clean/cleaned/cleaning，它们都和clean是一个意思，因此采用nltk库对单词提取词干
* **数据标准化：**把某个特征的所有样本转换成均值为0，方差为1。
* **数据归一化：**把某个特征的所有样本取值限定在规定范围内（一般为[-1,1]或者[0,1]）。
* **数据缺值处理：**对于缺失的特征数据，进行数据填补，一般填补的方法有：均值，中位数，众数填补等。
* **数据离群点处理：**删除离群点数据。
* **数据类型转换：**如果数据的特征不是数值型特征，则需要转换为数值型。

### 特征工程

特征是数据中抽取出来的对结果预测有用的信息，可以是文本或者数据。特征工程是使用专业背景知识和技巧处理数据，使得特征能在机器学习算法上发挥更好的作用的过程。过程包含了特征提取、特征构建、特征选择等模块。

特征工程的目的是筛选出更好的特征，获取更好的训练数据。因为好的特征具有更强的灵活性，可以用简单的模型做训练，更可以得到优秀的结果。“工欲善其事，必先利其器”，特征工程可以理解为利其器的过程。

### 模型搭建与训练

模型采用多种机器学习算法，通过最终投票的方式选取最终的排序结果，其中机器学习算法包括XGBoost Linear Model，RandomForest，SVR等。

### 数据导入模块

运行程序后从后台读入训练数据和测试数据。

### 模型导入模块

运行程序后从后台读入训练好的模型。

### 搜索模块

* 用户选择搜索关键字进行搜索；
* 模型通过计算，根据关键字得到对应的文本相关度，并按相关度降序显示搜索结果。

# 总体设计

## 类图



### 类图综述

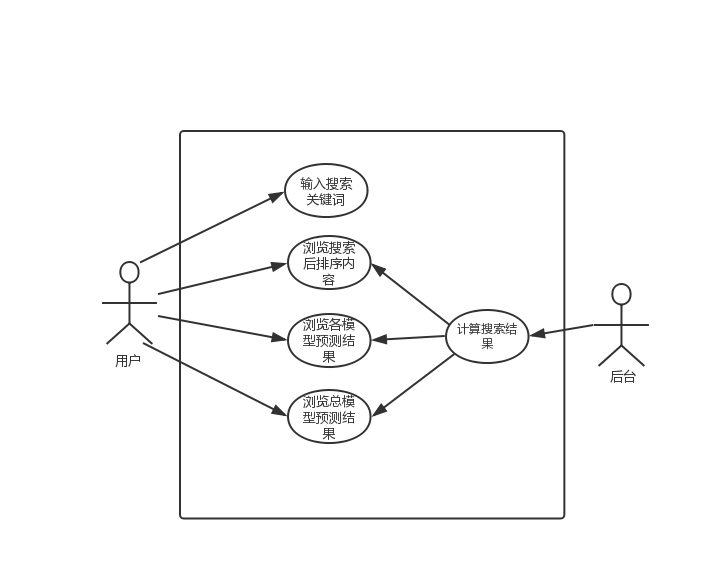
类图描述了本项目中所包含的类及类间关系，类图从整体上反映了用类描述的系统结构。本项目主要包括了搜索类，包括数据类，模型类，以及后端类，包括数据处理类，模型训练类，前后端接口类。并描述了他们之间的关联、依赖、接口关系。

### 类描述

|  |  |  |
| --- | --- | --- |
| **类** | **属性** | |
| Search | 关键字 | InputKeyword |
| **方法** | |
| 输入关键字 | InputKeyword() |
| 点击搜索 | ClickSearch() |
| **类** | **属性** | |
| Data | 数据集 | Dataset |
| **方法** | |
| 加载数据集 | LoadData() |
| **类** | **属性** | |
| Model | 模型 | Model |
| **方法** | |
| 加载模型 | LoadData() |
| **类** | **属性** | |
| BackEnd | 数据集 | DataSet |
| 模型 | Model |
| **方法** | |
| 返回数据 | ReturnModelData() |
| 返回模型 | ReturnModel() |
| **类** | **属性** | |
| DataProcess | 数据集 | DataSet |
| 训练集 | X\_Train |
| 训练集标签 | Y\_train |
| 测试集 | X\_test |
| 测试集标签 | Y\_test |
| **方法** | |
| 数据集初始化 | DataInit() |
| 数据清洗 | DataClean() |
| 类 | 属性 | |
| TrainModel | 模型 | Model |
| 方法 | |
| 加载数据 | LoadData() |
| 返回模型 | ReturnModel() |
| 训练模型 | Train() |
| 类 | 属性 | |
| Connector | 数据关键字 | Keyword |
| 方法 | |
| 获取输入 | GetInput() |
| 返回模型结果 | ReturnModelData() |

## 用例图

### 用例图

****

### 参与者

本系统的参与者主要指使用该系统观察预测crowdflower词条匹配数据的人群，“后台”是一种抽象，完成由系统自动处理的功能，譬如数据和训练模型导入的功能，反馈给前端不同词条的计算结果，将前端展示给用户。

### 用例图描述

|  |
| --- |
| **用例编号：**UC01  **用例名称：**搜索内容  **参与者：**用户  **简要说明：**搜索关键词。  **基本事件流：**用户在关键字List中选择想要了解的关键字  **扩展事件流：**无。  **关系描述：**无。  **前置条件：** 预测数据集已成功导入系统。  **后置条件：**无。  **限制：**输入信息仅包括List中已有的关键字item。 |

|  |
| --- |
| **用例编号：**UC02  **用例名称：**浏览相关性评分  **参与者：**用户  **简要说明：**用户可查看各搜索结果的相关性评分。  **基本事件流：**用户在选择关键字后，可在描述List控件中查看到该关键字下，所处理的内容，且呈item项显示。  **扩展事件流：** 系统计算保留该搜索结果下的各模型的相关性评分。  **关系描述：**无。  **前置条件：** 预测模型正常导入，系统正常运行。  **后置条件：**无。  **限制：** 无 |

|  |
| --- |
| **用例编号：**UC03  **用例名称：**浏览详情页信息  **参与者：**用户  **简要说明：**用户在某一搜索结果下可查看其各预测模型下的预测值。  **基本事件流：**用户在搜索结果页面，选择搜索结果列表里的某一项搜索结果，可跳转到详情页查看到每个预测模型具体数值以及该项搜索结果的具体信息。  **扩展事件流：无**  **关系描述：**无。  **前置条件：** 系统正常运行。  **后置条件：**无。  **限制：**搜索结果页面成功呈现出计算相关度评分结果。 |

# 接口设计

## 外部接口设计

## 内部接口设计

|  |  |  |
| --- | --- | --- |
| 1. 界面描述 | 界面初稿 | 界面信息 |
| **首页：**顶部由小组logo，主体部分是两个ListCtrl控件，左边是一系列的已有关键字选项 ，右侧是对应的搜索结果 |  | 色号为（255.255，225）的米黄色背景 |
| 提示窗口信息用的下标提示栏 |
| 关键字ListCtrl控件 |
| 搜索结果ListCtrl控件 |
| 可选关键字 |
| 相关度评分 |
| 搜索结果Title |
| 搜索结果描述 |
| **评分详情页：**显示每一个搜索结果下的具体计算过程吗，以及构建的各模型下的具体评分和相关度。 |  | 色号为（225.225，213）的米黄色背景 |
| 三个预测模的图标表示 |
| 三个预测模型下的ACC指和relevance值 |
| 该项搜索结果的具体信息扩展显示 |

|  |  |  |
| --- | --- | --- |
| 文件名 | 路径 | 作用 |
| DataPreprocesss.py | ModelSystem/DataPreprocesss/ | 对原始数据集train.csv和test.csv进行预处理 |
| GenerateFeature.py | ModelSystem/Features/cotfidf/ | 生成交叉TF-TDF特征 |
| GenerateFeature.py | ModelSystem/Features/count / | 生成ngram计数特征 |
| GenerateFeature.py | ModelSystem/Features/distance/ | 生成距离特征（Jaccard距离和Dice距离） |
| GenerateFeature.py | ModelSystem/Features/ngram / | 对预处理之后的文本进行ngram分词，并存储分词结果 |
| GenerateFeature.py | ModelSystem/Features/tfidf/ | 生成TF-IDF特征 |
| GenerateFeature.py | ModelSystem/Features/group1/ | 生成Group1特征 |
| GenerateFeature.py | ModelSystem/Features/group2/ | 生成Group2特征 |
| GenerateFeature.py | ModelSystem/Features/group4/ | 生成Group4特征 |
| GenerateFeature.py | ModelSystem/Features/group5/ | 生成Group5特征 |
| CombineFeatures.py | ModelSystem/Features/ | 选定一部份特征，将这些特征组合成模型的训练数据x\_train |
| kappa.py | ModelSystem/Models/ | 使用kappa系数来对模型的预测性能进行评估 |

# 模型开发过程设计

## 数据预处理

### 功能

由于数据集的文本中混有错别词、html符号等错误信息，需要对数据集进行清洗，去除数据中多余、谬误或残缺的部分，以保证后续工作的正常展开。

### 输入

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **输入项目** | **媒介体** | **数据类型** | **说明** | **举例** |
| 原始数据 | N/A | String | 数据来源于Kaggle数据集 | 输入：”This movie is goood.” |

### 输出

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **输出项目** | **媒介体** | **数据类型** | **说明** | **举例** |
| 预处理后数据 | N/A | String |  | 输出：”This movie is good.” |

### 具体实现

**删除HTML标签**：由于数据集是从网页中获取，会有很多干扰的无用html标签存在，因此采用bs4库的BeautifulSoup函数来清除它们。

**同义词替换：**数据中有很多单词其实是同样的意思，因此我们把这些同义词都替换成统一的一个，如：将’refrigerator’’fridge’’freezer’全部替换成‘fridge’。首先构建同义词表，再将对应的单词替换成统一的标准单词。

**其他替换：**数据中有一些组合词是分开的，不便于训练和处理，比如‘16 gb’‘32 g’，因此我们将他们全部改成合并的，如‘16gb’‘32g’。采用re库的sub函数来实现此替换。

**词干提取：**英文中有很多变形，比如clean/cleaned/cleaning，它们都和clean是一个意思，因此采用nltk库的stem方法对单词提取词干。

## 特征工程

### 功能

特征是数据中抽取出来的对结果预测有用的信息，可以是文本或者数据。特征工程是使用专业背景知识和技巧处理数据，使得特征能在机器学习算法上发挥更好的作用的过程。过程包含了特征提取、特征构建、特征选择等模块。

特征工程的目的是筛选出更好的特征，获取更好的训练数据。因为好的特征具有更强的灵活性，可以用简单的模型做训练，更可以得到优秀的结果。

### 输入

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **输入项目** | **媒介体** | **数据类型** | **说明** | **举例** |
| 预处理后的数据 | N/A | String |  | 输入：”This movie is good.” |

### 输出

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **输出项目** | **媒介体** | **数据类型** | **说明** | **举例** |
| 提取的数据特征 | N/A | N/A |  | 文本的tf-idf或文本经word2vec后的词向量矩阵 |
| 构建的特征 | N/A | N/A |  |  |

### 具体实现

记号说明：

原数据集具有三个原始特征query,product\_title, prodict\_description, 对于位于数据集中的第i个样本的这三个特征，分别记为qi, ti ,di

将对x进行y元文法分词后所得的分词列表记为ngram(x,y)

以下为提取的特征：

* **ngram分词个数，共9个**

ngram(ti,1)中分词的个数

ngram(ti,2)中分词的个数

ngram(ti,3)中分词的个数

ngram(qi,1)中分词的个数

ngram(qi,2)中分词的个数

ngram(qi,3)中分词的个数

ngram(di,1)中分词的个数

ngram(di,2)中分词的个数

ngram(di,3)中分词的个数

* **ngram唯一分词的个数及其比例，共9个**

对ti进行一元文法分词后，只出现一次的分词的个数及其在整个分词集所占的比例

对ti进行二元文法分词后，只出现一次的分词的个数及其在整个分词集所占的比例

对ti进行三元文法分词后，只出现一次的分词的个数及其在整个分词集所占的比例

对qi进行一元文法分词后，只出现一次的分词的个数及其在整个分词集所占的比例

对qi进行二元文法分词后，只出现一次的分词的个数及其在整个分词集所占的比例

对qi进行三元文法分词后，只出现一次的分词的个数及其在整个分词集所占的比例

对di进行一元文法分词后，只出现一次的分词的个数及其在整个分词集所占的比例

对di进行二元文法分词后，只出现一次的分词的个数及其在整个分词集所占的比例

对di进行三元文法分词后，只出现一次的分词的个数及其在整个分词集所占的比例

* **描述丢失指示，共1个**

01指示器，表示di数据是否缺失

* **TF-IDF 原始特征，共3个**

qi, ti, di各自的TF-IDF（ngram\_range=[1,3]）

* **基于TF-IDF的余弦相似度，共3个**

ti, qi的TF-IDF的余弦相似度

ti, di的TF-IDF的余弦相似度

qi, di的TF-IDF的余弦相似度

* **TD-IDF SVD，共3个**

对qi, ti, di各自的TF-IDF进行SVD降维至100维后的特征

* **Jaccard系数，共19个**

记J(A,B)为集合A,B的Jaccard系数，其中

https://gss3.bdstatic.com/-Po3dSag_xI4khGkpoWK1HF6hhy/baike/s%3D115/sign=7f5efa23379b033b2888f8db20ce3620/203fb80e7bec54e79da5fd39ba389b504fc26a48.jpg，生成的特征有：

J(ngram(qi,1), ngram(di, 1))

J(ngram(qi,1), ngram(di, 2))

J(ngram(qi,1), ngram(di, 3))

J(ngram(qi,1), ngram(ti, 1))

J(ngram(qi,1), ngram(ti, 2))

J(ngram(qi,1), ngram(ti, 3))

J(ngram(ti,1), ngram(di, 1))

J(ngram(ti,1), ngram(di, 2))

J(ngram(ti,1), ngram(di, 3))

J(ngram(ti,1), ngram(qi, 2))

J(ngram(ti,1), ngram(qi, 3))

J(ngram(qi,2), ngram(di, 2))

J(ngram(qi,2), ngram(di, 3))

J(ngram(ti,2), ngram(di, 2))

J(ngram(ti,2), ngram(di, 3))

J(ngram(di,2), ngram(qi, 3))

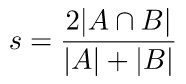
J(ngram(di,2), ngram(ti, 3))

J(ngram(qi,3), ngram(di, 3))

J(ngram(qi,3), ngram(qi, 3)

* **Dice距离，共19个**

记S(A,B)为集合A,B的Dice距离，其中

****，生成的特征有

S(ngram(qi,1), ngram(di, 1))

S(ngram(qi,1), ngram(di, 2))

S(ngram(qi,1), ngram(di, 3))

S(ngram(qi,1), ngram(ti, 1))

S(ngram(qi,1), ngram(ti, 2))

S(ngram(qi,1), ngram(ti, 3))

S(ngram(ti,1), ngram(di, 1))

S(ngram(ti,1), ngram(di, 2))

S(ngram(ti,1), ngram(di, 3))

S(ngram(ti,1), ngram(qi, 2))

S(ngram(ti,1), ngram(qi, 3))

S(ngram(qi,2), ngram(di, 2))

S(ngram(qi,2), ngram(di, 3))

S(ngram(ti,2), ngram(di, 2))

S(ngram(ti,2), ngram(di, 3))

S(ngram(di,2), ngram(qi, 3))

S(ngram(di,2), ngram(ti, 3))

S(ngram(qi,3), ngram(di, 3))

S(ngram(qi,3), ngram(qi, 3)

* **交叉TF-IDF，共9个**

记号说明：

CO(A,B):设A=[”I”, ”love”, “China”]，B=[”Good”, “morning”]，则CO(A,B) = “I\_Good I\_morning love\_Good love\_moring China\_Good China\_morning”

TF-IDF(x):文段x对应的TF-TDF原始特征

生成的特征为

TF-IDF(CO(ngram(qi,1), ngram(di, 1)))

TF-IDF(CO(ngram(qi,1), ngram(di, 2)))

TF-IDF(CO(ngram(qi,1), ngram(ti, 1)))

TF-IDF(CO(ngram(qi,1), ngram(ti, 2)))

TF-IDF(CO(ngram(ti,1), ngram(di, 1)))

TF-IDF(CO(ngram(ti,1), ngram(di, 2)))

TF-IDF(CO(ngram(ti,1), ngram(qi, 2)))

TF-IDF(CO(ngram(qi,2), ngram(di, 2)))

TF-IDF(CO(ngram(ti,2), ngram(di, 2)))

* **Group 1，共6个**

出现在ti中的qi的词语个数

出现在ti中的qi的词语占qi词语个数的比例

qi到ti的压缩距离（lzma）

qi到ti的基于语句的编辑距离（edit distance）

qi到ti的基于单词的最大编辑距离的平均值

* **Group 2，共6个**

记拓展之后的qi为exqi，生成的特征有：

出现在ti中的exqi的词语个数

出现在ti中的exqi的词语占exqi词语个数的比例

exqi到ti的压缩距离（lzma）

exqi到ti的基于语句的编辑距离（edit distance）

exqi到ti的基于单词的最大编辑距离的平均值

* **Group 4，共1个**

( Freq(qi) + A ) / ( Freq(ti) + B ) / ( len(qi))

其中A，B为常数, Freq(x)表示文段中各个字母出现的频数，len(x)表示文段x的长度

* **Group 5, 共4个**

说明：我们使用了Google提供的已经训练好的word2vec模型（GoogleNews-vectors-negative300.bin）

记BoW(x)为文段x的BoW特征，SVD(x,n)为将特征向量x至n维后的特征向量，tSNE（x,n）为将特征向量x降至n维后的特征向量

提取的特征有：

ti与qi的word2vec相似度

ti与qi的基于word2vec的距离（取文段中每个word的向量的平均作为该文段的向量）

ti与qi的基于word2vec的余弦相似度

tSNE ( SVD( BoW( ti ), 200) , 2)

tSNE ( SVD( TF-IDF( ti ), 200) , 2)

## 模型搭建

### 功能

根据数据集的特征和实际情况，对输入数据进行编码，转化为计算机能识别的矩阵。同时，构建相适应的模型，如各种分类器，神经网络等。

### 输入

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **输入项目** | **媒介体** | **数据类型** | **说明** | **举例** |
| 提取的数据特征 | N/A | N/A |  | 文本的tf-idf或文本经word2vec后的词向量矩阵 |

### 输出

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **输出项目** | **媒介体** | **数据类型** | **说明** | **举例** |
| 模型 | N/A | N/A | 包含随机森林、支持向量机、XGBoost等多种模型 |  |

### 具体实现

通过输入特征X\_train进行模型的训练，并将训练好的模型进行保存，在使用时进行载入调用，使用训练好的模型进行预测。

具体的调参方法采用hyperopt进行模型参数的调整，通过给定各个参数的范围使用hyperopt进行参数的自动优化。

具体算法：

1. 支持向量机

支持向量机就是使用一条直线（二维）或超平面（多维）将数据分成两类，同时保证离超平面最近的点与超平面的间隔尽可能小。

支持向量回归（SVR）：传统的回归模型是计算模型输出f(x)与真实值输出y之间的差别来计算损失，而支持向量回归有个容忍偏差ϵ，即仅当f(x)与y差值大于ϵ时才计算损失。

1. 随机森林

决策树是一种基本的分类方法，当然也可以用于回归。我们一般只讨论用于分类的决策树。决策树模型呈树形结构。在分类问题中，表示基于特征对实例进行分类的过程，它可以认为是if-then规则的集合。在决策树的结构中，每一个实例都被一条路径或者一条规则所覆盖。通常决策树学习包括三个步骤：特征选择、决策树的生成和决策树的修剪

在机器学习中，随机森林是一个包含多个决策树的分类器，并且其输出的类别是由个别树输出的类别的众数而定。利用相同的训练数搭建多个独立的分类模型，然后通过投票的方式，以少数服从多数的原则作出最终的分类决策。

1. XGBoost

xgboost是大规模并行boosted tree的工具，它是目前最快最好的开源boosted tree工具包，比常见的工具包快10倍以上。在数据科学方面，有大量kaggle选手选用它进行数据挖掘比赛，其中包括两个以上kaggle比赛的夺冠方案。在工业界规模方面，xgboost的分布式版本有广泛的可移植性，支持在YARN, MPI, Sungrid Engine等各个平台上面运行，并且保留了单机并行版本的各种优化，使得它可以很好地解决于工业界规模的问题。

1. 感知机

感知机是二类分类的线性分类模型，其输入为实例的特征向量，输出为实例的类别。感知机对应于输入空间（特征空间）中将实例划分为正负两类的分离超平面。感知机是一种线性分类模型。MLP又名多层感知机，也叫人工神经网络（ANN，Artificial Neural Network），除了输入输出层，它中间可以有多个隐藏层，如果没有隐藏层即可解决线性可划分的数据问题。

1. Auto-Sklearn

auto-sklearn是一个自动化机器学习的工具包,其基于sklearn编写。Auto-Sklearn主要基于sklearn机器学习库，使用方法也与之类似，这让熟悉sklearn的开发者很容易切换到Auto-Sklearn。在模型方面，除了sklearn提供的机器学习模型，还加入了xgboost算法支持；在框架整体调优方面，使用了贝叶斯优化。

1. 线性回归

线性回归(Linear Regression)是利用数理统计中回归分析，来确定两种或两种以上变量间相互依赖的定量关系的一种统计分析方法

线性回归利用称为线性回归方程的最小平方函数对一个或多个自变量和因变量之间关系进行建模。 这种函数是一个或多个称为回归系数的模型参数的线性组合。只有一个自变量的情况称为简单回归,大于一个自变量情况的叫做多元回归。

1. 岭回归

岭回归是一种专用于共线性数据分析的有偏估计回归方法，实质上是一种改良的最小二乘估计法，通过放弃最小二乘法的无偏性，以损失部分信息、降低精度为代价获得回归系数更为符合实际、更可靠的回归方法，对病态数据的拟合要强于最小二乘法。。

# 展示系统设计

## 数据导入模块

### 功能

在搜索功能启用之前，加载已有的模型数据。

### 输入

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **输入项目** | **媒介体** | **数据类型** | **说明** | **举例** |
| 已经评分完毕的数据 | N/A | 二进制 | 二进制文件，数据来源于测试集 |  |

* 1. **匹配结果模块**

### 功能

根据用户输入的关键字，使用训练好的模型来对数据进行分析、计算，再根据计算的相关度对资料库的每条搜索结果降序展示给用户。

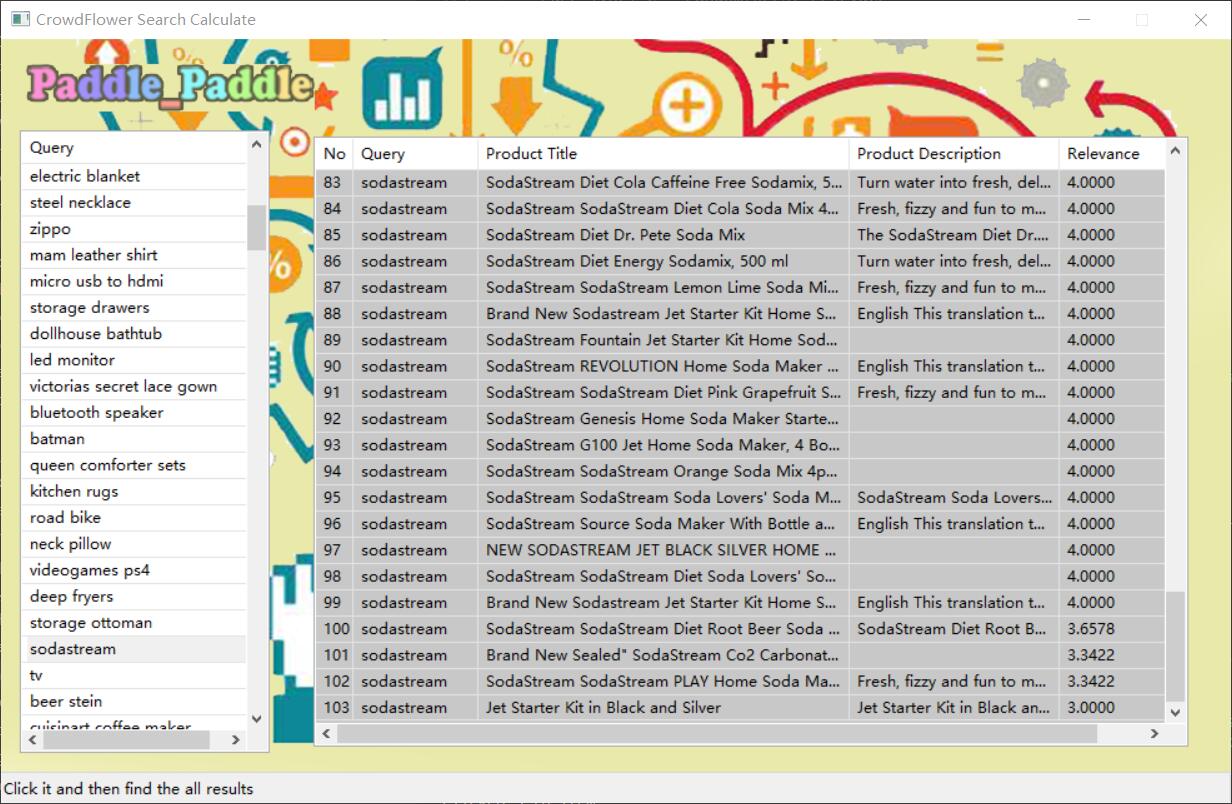
### 输入

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **输入项目** | **媒介体** | **数据类型** | **说明** | **举例** |
| 关键字模块传来的关键字选项内容 | List列表 | 大小写字母 / 数字 / 空格 | 只限于列表内的关键字 | 无 |

### 输出

输出可查看到该关键字检索下的crowdflower数据集范围内的相关检索内容并由高到低排序显示及其对应的拟合评分。

示例：



* 1. **评分详情模块**



### 功能

向用户展示每个综合评分下的各个预测模型的具体评分和相关度，并扩展显示该项搜索结果下的各项全部信息（Query、Title、Description），以及最终评分。

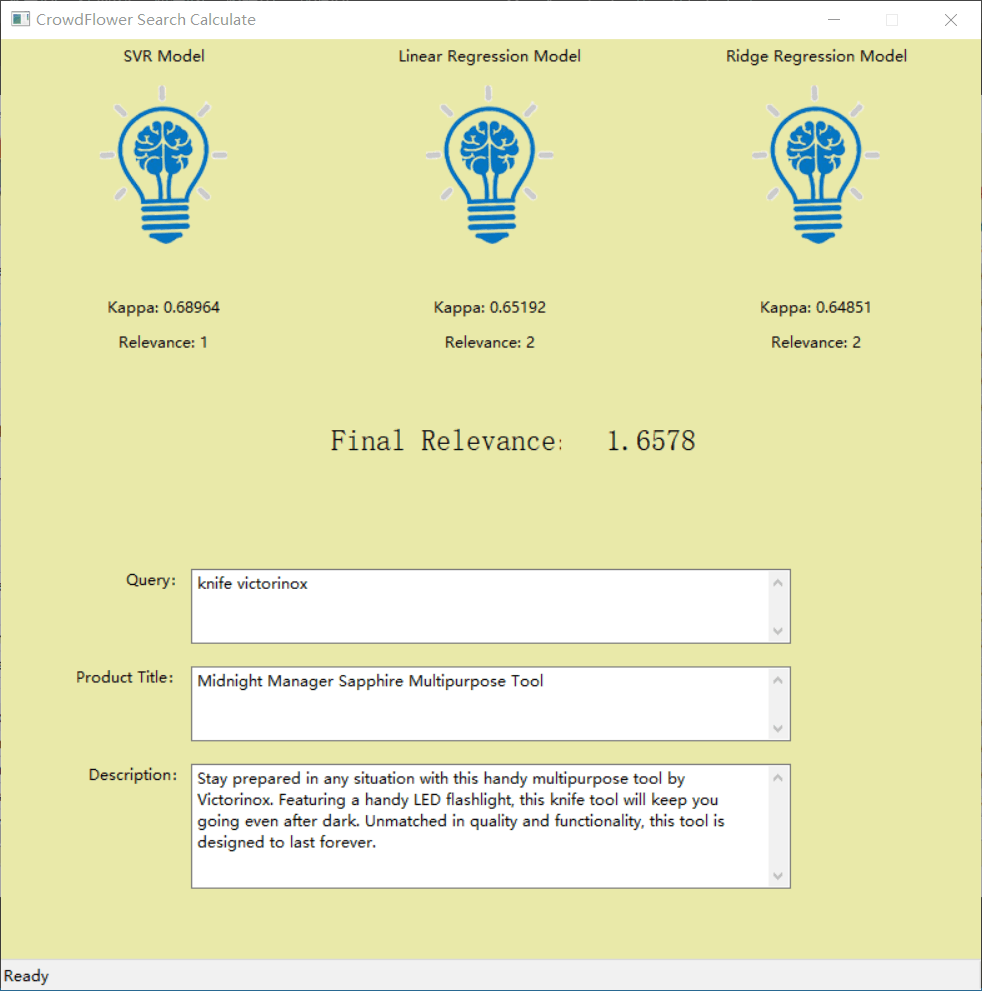
### 输入

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **输入项目** | **媒介体** | **数据类型** | **说明** | **举例** |
| 点击匹配界面的关键字项 | List列表 | 矩阵型的List数据 | 只限于列表内已有的item |  |

### 输出

显示各个模型下的具体评分和相关度，最终评分Final Relevance处于界面中部位置输出，并在其下面分三行horizon布局并显示该项搜索结果下的各项全部信息（Query、Title、Description）。

示例：



### 程序逻辑

|  |  |
| --- | --- |
| **描述** | **UML图** |
| **活动图综述**：该活动图描述了“搜索用户或关键词”的过程，涉及用户和后台两个对象。  **参与者对象描述**： “用户”和“后台”是参与者，用户能实现搜索用户或关键词操作。  **状态描述**：用户在搜索时，先后进行操作：输入关键字、点击搜索后即可看到搜索信息，默认为动态搜索，点击按钮可切换为用户搜索。  **转换描述**：在“搜索”活动中，有两个分支控制：第一个是在用户输入关键字并点击搜索后，后台根据关键字检索，若存在与关键字匹配的动态信息和用户名，则返回信息；若没有匹配的信息，则返回一个空对象；第二个是在用户在底部继续下拉时，后台会根据搜索的关键字及页面已显示数据条数的信息返回新的数据。 | C:\Users\lenovo\Desktop\临时.png |

# 提示信息设计

主要写下statusBar里面的信息

|  |  |  |
| --- | --- | --- |
| 触发条件 | 提示信息 | 示例 |
| 默认 | Ready |  |
| 鼠标移动在关键字列表里（不包括title和滑动栏） | Click it and then find the all results |  |
| 鼠标移动在匹配列表里（不包括title和滑动栏） | Click it and then find the details of the calculation |  |
| 第二个Page页面的默认（详情页） | Ready |  |