# 商标平台详细设计文档

## ？、功能模块设计

### ？.1 数据预处理模块

#### ?.1.1 历史数据转储子模块

#### ?.1.2 历史数据转储子模块

### ？.2 分类模型训练模块

#### ?.2.1 近似商标采集子模块

#### ?.2.2 特征数据转换子模块

#### ?.2.3 模型训练子模块

### ？.3 近似商标检索接口



#### ?.3.1 近似商标检索子模块

* 输入： 1、商标名

2、查询的大类编号列表

* 输出：如《商标AI接口文档》中定义的大类近似商标检索结果实体类，以输入数据中涵盖的大类为准，每个大类返回一个实体类结果。
* 处理流程：

1. 解析输入商标名，拆分为中文、英文和数字三部分，用中文转换得到拼音。
2. 获取（部分）同音历史商标名：

2.1、使用1中转换的拼音的长度inp\_len（按一个字的读音为1算），计算一个同音商标名的最小同音字数lowb，规则为：

* + 1. 当inp\_len 2时，最小同音字数lowb = inp\_len
    2. 当inp\_len 3 时，最小同音字数lowb = max ( inp\_len \* 0.5 , 2 )

待平台方确定具体的不同长度下的同音限制后再修改

2.2、用lowb对1中转换的拼音计算排列组合，使用得出的组合从redis数据库中查询各个大类下满足这样的读音组合的商标名。

1. 同音商标名过滤筛选

比较输入商标的读音序列inp\_py和历史商标的读音序列his\_py，参考《查询商标近似需求文档》中的有关定义，作如下步骤的过滤

* + 1. 当输入商标字数3时，按乱序查询，直接比较看输入商标的读音序列inp\_py和历史商标的读音序列his\_py有几个读音是相同的，记作同音字数comm，转iii；
    2. 当输入商标字数3时，只按正序查询，对输入商标的读音序列inp\_py和历史商标的读音序列his\_py，计算两者的最大正序匹配长度，记作同音字数comm，转iii；
    3. 阈值过滤：若历史商标名的长度comm + 4，即比同音部分还长4个字以上，则过滤这条历史数据。（待平台方确定阈值后调整）
    4. 余下数据通过过滤，进入4

1. 同音商标名和输入商标名计算特征值
2. redis查询同音商标名注册的商品项
3. 调用特征数据转换子模块，将4步计算的特征数据交付转换、计算是否通过
4. 接收调用的结果（已经在子模块中封装为实体类），返回结果（给调用接口）

#### ?.3.2 特征数据转换子模块

* 输入：

由近似商标检索模块筛选过滤得到的近似历史商标与输入商标计算的多项相似度数据，以及近似商标的一些数据包括：输入商标名、近似商标名、近似商标注册号、检索的大类、近似商标注册的商品项，用于构造结果类。

* 输出：

如《商标AI接口文档》中定义的大类近似商标检索结果实体类，以输入数据中涵盖的大类为准，每个大类返回一个实体类结果。

* 步骤：

1、 解析输入数据，取出输入数据与近似商标计算的特征数据，做libsvn格式的压缩，每个相似度数据被描述成 <index:value> 形式的组合， 一条数据最后被描述如下一行数据：

y index1:value1 index2:value2 …. indexn:valuen

其中y填写任意值，indexi是第i个特征的下标，valuei是第i个特征的取值。上述数据保存为输入数据文件。

2、调用模型分类子模块，对1步保存的输入数据进行计算，得到每条数据（实际上也是每个历史商标和输入商标的比较结果）的近似通过率。

3、将第2步调用返回的计算结果、连同模块的输入数据，封装为《商标AI接口文档》中定义的大类近似商标检索结果实体类。作如下操作：

3.1、输入模块的每个近似名字，连同其注册号、近似度，封装为近似名字实体类，调用近似标签匹配模块，对每个近似商标的近似标签（名字完全相同、名字部分相同前面加字等）做判断，结果一并封装在实体类中。

3.2、根据输入的每个近似商标注册的商品项，封装/更新对应的商品项及成功率列表实体类，使每个实体类的对应注册成功率是所有计算结果中的最小值。

3.3、将3.1、3.2封装的实体类结果按大类归类整理，封装为大类近似商标检索结果实体类，并补上每个大类中，近似商标都没有注册到的商品项对应的实体类，标记其通过率为100。

4、结果实体类返回输出到上层调用模块。

#### ?.3.3 模型分类子模块

* 输入：输入数据文件、模型文件
* 输出：每行输入数据输出一个通过概率
* 步骤：

1、读取配置文件，确定并加载xgboost分类模型

2、读取输入数据文件/矩阵，转化为xgboost的DMatrix格式

3、使用模型对输入数据预测计算（0-1分类），得到m\*2的概率矩阵，m为输入数据的行数

4、遍历计算结果，以分类为1作为近似判断通过。将分类为1的概率作为通过概率，结果返回上层调用模块