BIA API

BIA是Bethune Intelligent Analysis（白求恩智能分析）的简写，BIA使用机器学习来实现智能分析任务，本文档是BIA的API开发说明，BIA的package是com.enmotech.bethune.bia。需要注意的是，因BIA是基于WEKA开发的，所以在项目中除了加入bia-1.0.jar以外，必须加入weka.jar，否则BIA无法正常工作。WEKA的maven配置如下：

|  |
| --- |
| <**dependency**>  <**groupId**>nz.ac.waikato.cms.weka</**groupId**>  <**artifactId**>weka-stable</**artifactId**>  <**version**>3.8.1</**version**> </**dependency**>  <**dependency**>  <**groupId**>nz.ac.waikato.cms.weka</**groupId**>  <**artifactId**>timeseriesForecasting</**artifactId**>  <**version**>1.1.25</**version**> </**dependency**>  <**dependency**>  <**groupId**>com.github.servicenow.stl4j</**groupId**>  <**artifactId**>stl-decomp-4j</**artifactId**>  <**version**>1.0.2</**version**> </**dependency**> |

BIA使用说明：BIA的实现在com.enmotech.bethune.bia.analysis下面，这个包下有三个类，如表所示：

|  |  |  |  |
| --- | --- | --- | --- |
| 类名称 | 类类型 | 父类 | 功能说明 |
| BethuneAnalysis | interface | 无 | 提供最基础的接口 |
| IntelligentBethune | abstract | BethuneAnalysis | 实现基础的功能，构建WEKA需要的数据结构等 |
| TimeSeriesForecast | Concret class | IntelligentBethune | 时间序列预测及其异常检查 |

TimeSeriesForecast方法及其参数说明：

|  |  |  |
| --- | --- | --- |
| 方法名称 | 方法类型 | 参数说明 |
| **public** TimeSeriesForecast(Map<String, List<Object>> dataMap, **double** threshold, **int** steps) | 构造方法 | ***dataMap***：时间序列，格式为{"timeFieldName":["2017", "2018", "2019"], "valueAttrName":[1232, 3240, 2343]}，Map的key可以任意指定，并且key的值可以是乱序的，但是必须确保timeFieldName和valueFieldName的List中的值一一对应，并且两个集合的长度必须相等，当前TimeSeriesForecast仅支持一个时间序列。  ***threshold*** ： 阀值，时间序列在预测中达到的某个值，如果为0，则在预测时不会去判断预测结果累计的和是否达到这个值。  ***steps*** ： 每一次预测的数量 |
| **public void** predict() **throws** Exception |  | 对时间序列进行预测。 |
| **public** Map<String, List<Object>> predictResults() **throws** Exception |  | 预测的结果，返回的数据结构和构造方法的dataMap一样的数据结构。  注：此方法必须在调用predict()方法后才能被调用，否则将抛出NullPointException异常 |
| **public int** getAlarmDays() |  | 返回预测结果累和达到threshold时的天数，所以threshold必须不能为0，否则返回0 |
| **public** Map<String, List<Object>> getSmoothedData() |  | 返回经过平滑处理的数据。 |
| **public** Map<String, List<Object>> outliers() **throws** Exception |  | 返回检测到的异常数据，此数据已经根据异常级别进行排序，越靠前说明异常级别越高。实例化TimeSeriesForecast后即可直接调用，无需执行predict()。 |

Exmaple:

|  |
| --- |
| **public class** ForecastTest **extends** TestCase {   @Test  **public void** test() **throws** IOException {    LinkedHashMap<String, List<Object>> dataMap = getDataFromCSV();  TimeSeriesForecast forecast = **new** TimeSeriesForecast(dataMap, 11111111, 5);  **try** {  Map<String, List<Object>> outlierMap = forecast.outliers();  System.***out***.println(**"--------------------------outliers----------------------------------"**);  System.***out***.println(outlierMap);  System.***out***.println(**"--------------------------smoothed data-----------------------------"**);  System.***out***.println(forecast.getSmoothedData());  forecast.predict();  **int** days = forecast.getAlarmDays();  Map<String, List<Object>> predictResults = forecast.predictResults();  System.***out***.println(**"the critical value is : "** + days);  System.***out***.println(**"--------------------------Predict Results--------------------------"**);  System.***out***.println(**"the predict results is:"**);  System.***out***.println(predictResults);  } **catch** (Exception e) {  e.printStackTrace();  }  }   **private** LinkedHashMap<String, List<Object>> getDataFromCSV() **throws** IOException {  String filePath = **"../data/PD\_DB\_AWR\_DBTIME.csv"**;  CSVReader reader = **new** CSVReader(**new** FileReader(filePath));   String[] fields = **null**;  String[] values = **null**;   LinkedHashMap<String, List<Object>> dataMap = **new** LinkedHashMap<>();  List<Object> dataList = **new** ArrayList<Object>();   **while** ((values = reader.readNext()) != **null**) {   **if** (fields == **null** || fields.**length** <= 0) {  fields = values;  **for** (**int** i = 0; i < 3; i++) {  dataMap.put(fields[i], **new** ArrayList<>());  }  **continue**;  }   **if** (**"jydqdb2"**.equals(values[0])) **continue**;   **for** (**int** i = 0; i < 3; i++) {  dataMap.get(fields[i]).add(values[i]);  } *// dataList.add(values);* }  **return** dataMap;  } } |