DataSplitter

2019年3月18日

13:45

DataSpliter接口：

声明枚举类型 SplitterType；

声明void型无参方法 splitterData() ，可抛出 LibrecException异常

声明void型 （DataConvertor 参数）方法setDataConvertor(DataConvertor dataConvertor)

引用包 顺序访问稀疏矩阵 （SequentialAccessSparseMatrix ）方法

|  |  |  |
| --- | --- | --- |
| getTrainData(); | getTestData(); | getValidData(); |

声明布尔型无参方法 nextFold()；

AbstractDataSplitter

2019年3月18日

18:58

声明了抽象类 AbstractDataSplitter ：

该抽象类继承于类Configured 并实现了接口DataSplitter

声明了 DataConvertor 型变量dataConvertor的 set 方法 setDataConvertor(DataConvertor dataConvertor)

声明了 SequentialAccessSparseMatrix 型无参方法 getTrainData（）；getTestData()；getValidData()；分别返回了在本抽象类AbstractDataSplitter 的 受保护变量 trainMatrix；testMatrix; validationMatrix;

声明了 SequentialAccessSparseMatrix 型变量 preferenceMatrix 的 set 方法 setPreferenceMatrix(SequentialAccessSparseMatrix preferenceMatrix)

重写了布尔型方法 nextFold()

判断 若 SequentialAccessSparseMatrix 型Linkedlist 变量 assignMatrix为空

则使assignMatrix为一个新的Linkedlist

返回true

若 assignMatrix 长度为<=0

返回false

若 assignMatrix 长度大于0

则对其内容进行遍历

若 assignMatrix的poll方法获取的 matrixEntry.row(), matrixEntry.column() 为真

则调用 trainMatrix 变量的setAtColumnPosition方法，并传入值

matrixEntry.row(), matrixEntry.columnPosition(), 0.0D

否则

调用 testMatrix 变量的setAtColumnPosition方法，并传入值

matrixEntry.row(), matrixEntry.columnPosition(), 0.0D

并分别调用trainMatrix 和 testMatrix 的 reshape()方法

返回true

GivenNDataSplitter

2019年3月18日

20:06

定义了类 GivenNDataSplitter

该类继承自 AbstractDataSplitter 类

声明了无参构造GivenNDataSplitter()

有参构造GivenNDataSplitter( DataConvertor dataConvertor, Configuration conf)

变量分别为为 dataConvertor； conf 并赋值

定义了void型方法splitData() 可抛出LibrecExceptio 错误

该方法对数据进行筛选

若preferenceMatrix 变量为空

则 调用 dataConvertor 的 getPreferenceMatrix(conf) 为其赋值

若dataConvertor 没有实例化 且StringUtils 与conf.get("data.column.format"), "UIRT") 方法相等

则调用dataConvertor的getDatetimeMatrix()方法为datetimeMatrix赋值

并调用conf.get("data.splitter.givenn")判断其String 字符串

若为“user” “item”"userdate" " itemdate" 则分别尝试调用相应的

getGivenNByUser

getGivenNByItem

getGivenNByUserDate

getGivenNByItemDate

方法，并传入Integer调用parseInt()方法，其值为(conf.get("data.splitter.givenn.n")) 参数

然后分别实现了

getGivenNByUser(int numGiven);

getGivenNByItem(int numGiven);

getGivenNByUserDate(int numGiven);

getGivenNByItemDate(int numGiven);

四个方法

其中getGivenNByUser(int numGiven);

//若nunGiven>0

//从nunSize=0开始对preferenceMatrix的rowSize进行遍历

//若 preferenceMatrix的row(rowIndex).getNumEntries()的值大于

对nunGiven判断大小

进一步preferenceMatrix调用rowSize()判断其大小，

对于preferenceMatrix进一步做row ，getNumEntries 调用取其值大小

对于条件符合的做 testMatrix的setAtColumnPosition方法调用

否则做 traintMatrix的setAtColumnPosition方法调用

接着进行trainMatrix和 testMatrix分别调用SequentialAccessSparseMatrix的resgape()方法

getGivenNByUserDate(int numGiven)；方法

调用了SequentialAccessSparseMatrix 类

并调用了preferenceMatrix 对象的rowSize(),row 方法

以SequentialSparseVector类 的getNumEntries方法 取值做判断

对itemRatingVector容器调用循环调用RatingContext型List的add方法

并调用Collections.sort()方法对itemRatingList进行处理

对于符合条件的itemRatingList的size()方法的值

获得其get().getItem()值传入testMatrix或trainMatrix对象的setAtColumnPosition方法

最后调用了SequentialAccessSparseMatrix 实例 对象testMatrix和trainMatrix的reshape方法

getGivenNByItem(int numGiven)；方法

调用了SequentialAccessSparseMatrix类的reshape方法

setAtRowPosition方法

columnSize；column；getNumEntries方法

getGivenNByItem(int numGiven)；方法

调用了SequentialAccessSparseMatrix类的reshape方法

setAtRowPosition方法

columnSize；column；getNumEntries方法

调用了Vector.VectorEntry对象的position方法

调用了SequentialSparseVector类，对其实现的SequentialAccessSparseMatrix类的reshape方法

调用了RatingContext类对其实现了List对象的get方法

GivenTestSetDataSplitter

2019年3月19日

12:34

* + 类GivenTestSetDataSplitter继承自AbstractDataSplitter
    - 声明了SequentialAccessSparseMatrix型私有变量preferenceMatrix
    - 声明了无参构造GivenTestSetDataSplitter()
    - 声明了有参构造GivenTestSetDataSplitter()
      * DataConvertor 型变量 convertor
      * Configuration 型变量 conf
    - 定义了 void 型方法 抛出 LiberException 异常
      * 声明了DataConvertor 型变量 testConvertor
      * 调用了Configuration类的get()方法赋值给 testConvertor
      * 调用了Configuration类的静态变量CONF\_DFS\_DATA\_DIR
      * 创建了TextDataConvertor型对象testConvertor
        + 判断testConvertor 内容分别实例化为TextDataConvertor型或ArffDataConvertor型
      * 尝试调用 DataConvertor 类的processData()方法
      * 调用了类MatrixEntry的set方法
      * 调用了 类SequentialAccessSparseMatrix的reshape()方法

KCVDataSplitter

2019年3月19日

12:35

* 1. 定义了类KCVDataSplitter 继承自AbstractDataSplitter
     1. 声明了SequentialAccessSparseMatrix型变量assignMatrix
     2. 声明了无参构造KCVDataSplitter()
     3. 定义了含参构造KCVDataSplitter((DataConvertor , Configuration )
     4. 定义了 void型 含参splitData(int)
        1. 调用了List类的sortList(List,boolean)方法
        2. 调用了DataConvertor类的
           1. getPreferenceMatrix(Configuration)方法
           2. getDatetimeMatrix()方法
        3. 判断了类ArffDataConvertor实例
        4. 调用了 Configuration类的get(String)方法
        5. 调用了ArrayList类的add()方法
        6. 调用了SequentialAccessSparseMatrix类的
           1. clone()方法
           2. getNumEntries()方法
           3. setAtColumnPosition()方法
           4. get()方法
           5. columnSize()方法
        7. 调用了HashBasedTable类的create()方法
     5. 定义了 void型 无参splitData()
        1. 调用了Configuration类的getint（String，int）
        2. 调用了本类含参splitData(int)
     6. 定义了List<SequentialAccessSparseMatrix> 型 无参方法getAssignMatrixList()

LOOCVDataSplitter

2019年3月19日

18:58

* 1. 定义了类LOOCVDataSplitter 继承自类AbstractDataSplitter，
     1. 声明了 SequentialAccessSparseMatrix 型变量 preferenceMatrix，datetimeMatrix
     2. 声明了KCVDataSplitter 型变量 kcv
     3. 定义了无参构造 LOOCVDataSplitter()
     4. 定义了含参构造 LOOCVDataSplitter(DataConvertor ，Configuration )分别为属性dataConvertor， conf 赋值
     5. 定义了枚举类 LOOCVType ，包含两个变量 LOOByUser, LOOByItem
     6. 定义了void型 无参方法splitData()，可抛出LibrecException 异常
        1. 在该方法中判断变量 preferenceMatrix 若为空，
           1. 则调用dataConvertor对象的getPreferenceMatrix(Configuration)方法,传入值为conf。

并判断dataConvertor属性若为类 ArffDataConvertor的实例且属性conf的get(String，String）方法的两个String 变量相等则调用dataConvertor对象的getDatetimeMatrix()方法为datetimeMatrix变量赋值

调用 对象conf 的get(String)方法，并根据其值分别调用

getLOOByUser(); getLOOByItems(); getLOOByUserDate(); getLooByItemsDate();

四个void型无参方法，

* 1. 定义了getLOOByUser();
     1. 为 trainMatrix 调用 SequentialAccessSparseMatrix类的SequentialAccessSparseMatrix(SequentialAccessSparseMatrix) 方法传入值为preferenceMatrix

使preferenceMatrix对象调用rowSize()方法，从0开始遍历其值

若preferenceMatrix对象的row(int)方法非空

循环为trainMatrix对象调用setAtColumnPosition（int，int，double）方法；dataTable对象调用put方法

对对象trainMatrix和 testMatrix分别调用SequentialAccessSparseMatrix的resgape()方法和SequentialAccessSparseMatrix(int , int)方法

* 1. 定义了getLOOByUserDate();
     1. 为 trainMatrix 调用 SequentialAccessSparseMatrix类的SequentialAccessSparseMatrix(SequentialAccessSparseMatrix) 方法传入值为preferenceMatrix

使preferenceMatrix对象调用rowSize()方法，从0开始遍历其值

为SequentialSparseVector 调用getNumEntries()，非空则

对容器VectorEntry遍历，为List<RatingContext>添加值

对trainMatrix对象调用setAtColumnPosition（int，int，double）方法；dataTable对象调用put方法

对对象trainMatrix和 testMatrix分别调用SequentialAccessSparseMatrix的resgape()方法和SequentialAccessSparseMatrix(int , int)方法

* 1. 定义了getLOOByItems()
     1. 为 trainMatrix 调用 SequentialAccessSparseMatrix类的SequentialAccessSparseMatrix(SequentialAccessSparseMatrix) 方法传入值为preferenceMatrix

使preferenceMatrix对象调用columnsize()方法，从0开始遍历其值

若preferenceMatrix对象的column(int)方法非空

循环为trainMatrix对象调用ssetAtRowPosition（int，int，double）方法；dataTable对象调用put方法

对对象trainMatrix和 testMatrix分别调用SequentialAccessSparseMatrix的resgape()方法和SequentialAccessSparseMatrix(int , int)方法

* 1. 定义了getLOOByItemsDate()
     1. 为 trainMatrix 调用 SequentialAccessSparseMatrix类的SequentialAccessSparseMatrix(SequentialAccessSparseMatrix) 方法传入值为preferenceMatrix

使preferenceMatrix对象调用cloumnsize()方法，从0开始遍历其值

为SequentialSparseVector 调用getNumEntries()，非空则

对容器VectorEntry遍历，为List<RatingContext>添加值

对trainMatrix对象调用setAtRowPosition（int，int，double）方法；dataTable对象调用put方法

对对象trainMatrix和 testMatrix分别调用SequentialAccessSparseMatrix的resgape()方法和SequentialAccessSparseMatrix(int , int)方法

RatioDataSplitter

2019年3月20日

18:10

* 1. 定义了类RationDataSplitter 继承自类AbstractDataSplitter，
     1. 定义了 SequentialAccessSparseMatrix 类私有变量datetimeMatrix
     2. 定义了无参构造RatioDataSplitter()
     3. 定义了含参构造RatioDataSplitter(DataConvertor ，Configuration )分别为属性dataConvertor， conf 赋值
     4. 定义了void型 无参方法splitData()，可抛出LibrecException 异常
        1. 在该方法中判断变量 preferenceMatrix 若为空
           1. ，则调用dataConvertor对象的getPreferenceMatrix(Configuration)方法,传入值为conf。

并判断dataConvertor属性若为类 ArffDataConvertor的实例且属性conf的get(String，String）方法的两个String 变量相等则调用dataConvertor对象的getDatetimeMatrix()方法为datetimeMatrix变量赋值

* 1. 调用 对象conf 的get(String)方法，并根据其值分别调用不同的 double 型 比率参数调用方法；分别为：
     1. void型方法 getRatioByRating(double )：

使用MatrixEntry 类对象遍历preferenceMatrix 对象

产生随机数分别调用 testMatrix和trainMatrix的etAtColumnPosition（int，int）方法，

遍历结束后分别调用 testMatrix和trainMatrix对象的reshape()方法

* 1. void型方法 getRatioByRatingDate(double )：
     1. 若double参数介于0~1，则 testMatrix和trainMatrix对象重新实例化SequentialAccessSparseMatrix（SequentialAccessSparseMatrix）

并重新为 List<RatingContext> 型 对象ratingContexts添加值

为小于ratingContexts长度比率的部分调用testMatrix对象的setAtColumnPosition(int,int,double),否则调用trainMatrix对象的setAtColumnPosition(int,int,double)方法

最后调用了 对象testMatrix和trainMatrix的reshape方法

* 1. void型方法 getFixedRatioByUser(double )：
     1. 若double参数介于0~1，则 testMatrix和trainMatrix对象重新实例化SequentialAccessSparseMatrix（SequentialAccessSparseMatrix）

从0开始遍历preferenceMatrix对象的rowSize()方法

为小于double参数的部分调用testMatrix对象的setAtColumnPosition(int,int,double),否则调用trainMatrix对象的setAtColumnPosition(int,int,double)方法

最后调用了 对象testMatrix和trainMatrix的reshape方法

* 1. void型方法getFixedRatioByUse（double）
     1. 若double参数介于0~1，则 testMatrix和trainMatrix对象重新实例化SequentialAccessSparseMatrix（SequentialAccessSparseMatrix）

从0开始遍历preferenceMatrix对象的rowSize()方法

对preferenceMatrix对象的row()比率进行Math方法取整操作，筛选小于1的值对其进行trainMatrix对象的setAtColumnPosition(int,int,double)操作

最后调用了 对象testMatrix和trainMatrix的reshape方法

* 1. void型方法 getRatioByUserDate(double )：
     1. 若double参数介于0~1，则 testMatrix和trainMatrix对象重新实例化

对preferenceMatrix对象rowSize()遍历

对List<RatingContext>型对象itemRatingList 添加值为其按比率调用testMatrix和trainMatrix对象的setAtColumnPosition(int,int,double)方法

最后调用了 对象testMatrix和trainMatrix的reshape方法

* 1. void型方法 getRatioByItem(double )：
     1. 若double参数介于0~1，则 testMatrix和trainMatrix对象重新实例化

遍历preferenceMatrix对象列数目，内层遍历列对象按比率调用testMatrix和trainMatrix对象的setAtColumnPosition(int,int,double)方法

最后调用了 对象testMatrix和trainMatrix的reshape方法

* 1. void型方法 getRatioByItemDate(double )：
     1. 若double参数介于0~1，则 testMatrix和trainMatrix对象重新实例化SequentialAccessSparseMatrix（SequentialAccessSparseMatrix）

并重新遍历userRatingVector容器为 List<RatingContext> 型 对象ratingContexts添加值，排序完成后

按比率调用testMatrix和trainMatrix对象的setAtColumnPosition(int,int,double)方法

最后调用了 对象testMatrix和trainMatrix的reshape方法

* 1. 定义了void型方法 getRatio(double , double )
     1. 筛选比率实例化 三个 SequentialAccessSparseMatrix 对象；遍历对象preferenceMatrix

随机数判断部分调用三个值的setAtColumnPosition(int,int,double)方法

最后调用了 三个值d reshape（）方法，完成测试、训练和验证效果