Java代码库主要包含以下：

* 分类代码库(cn.clickwise.clickad.classify)

Classifier:分类模型读取，及预测类别

Classifier()读取分类模型，主要是读取模型参数，

cate(String text) 读取未分类文本，转换成标准格式样本，预测类别

getSample(String text) 将普通文本转换成标准格式样本

docate(String sample\_line) 对标准格式样本使用模型进行类别预测

ClassifierUseText:指定分类算法、模型，对每一行中的某个字段进行类别预测，并将类别编号转换成真实的类别名称

* 用户特征查询

DataQuery:查询接口定义

State connect(Connection con):连接kv-value server

List<Record>queryUid(Key key):按照用户uid查询用户记录

List<Record>queryUidTop(Key key,int top):按照用户uid查询用户记录,限制返回结果的数目

DataStore:数据存取接口定义

State connect(Connection con): 连接kv-value server

State write2db(Record rec,int day): 将记录写入数据库中

State deleteExpired(TimeRange time)：删除过期的数据

DataQuery和DataStore有三种子类实现，分别使用ardb、cassandra、hbase进行存储

ArdbQuery:使用ardb作为底层存储继承抽象类DataQuery

ArdbStore:使用ardb作为底层存储继承抽象类DataStore

CassandraQuery:使用cassandra作为底层存储继承抽象类DataQuery

CassandraStore:使用cassandra作为底层存储继承抽象类DataStore

HBaseQuery:使用hbase作为底层存储继承抽象类DataQuery

HBaseStore:使用hbase作为底层存储继承抽象类DataStore

EasyQueryHandler:用户特征查询的http服务查询处理类

QueryEasyServer：用户特征查询的http服务类

DmpInquiry:dmp离线查询

fetchFromDmp(File keyFile,FilerecordFile,Dmpdmp)：

从某地区dmp获取keyFile里所有用户的记录

fetchFromAllDmps(int day)：

从所有地区dmp获取所有用户的记录

writeRecFile2DataStore(FilerecordFile,Connection con,Dmpdmp,int day)：

用户记录从文件写入kv存储

ScheduleTask:每天定时执行任务，从各地dmp平台获取用户数据，并将用户数据写入本地存储，以对外提供查询服务。

* JD股票

此部分的代码包含cn.clickwise.clickad.jd\_opinion中的内容和

cn.clickwise.web.bd.WordLinks、cn.clickwise.web.FieldCrawl、

cn.clickwise.liqi.str.app.TFIDFStatisticN等内容.

cn.clickwise.clickad.jd\_opinion. SEClickMR：从采集的nstat数据中，提取搜索词，并统计每个搜索词的pv数，结果按pv排序

cn.clickwise.clickad.jd\_opinion.ParseResult: 存储nstat数据一行的解析结果

cn.clickwise.clickad.jd\_opinion. SEUrlParse: 解析搜索连接，提取有用的信息,解析对象包括refer和url(host+uri)

cn.clickwise.web.bd.WordLinks:抓取关键的百度搜索结果中的链接信息

cn.clickwise.web.FieldCrawl：抓取链接(每一行中的某个字段)的内容，并将抓取内容放到 该行的结尾

cn.clickwise.liqi.str.app.TFIDFStatisticN: 统计一批文章(每一行一篇文章)中的关键词的得分情况，和采用该得分标准文章的得分。、

详细使用过程参加 JD\_STOCK文档。

* Radius数据采集:

此部分代码的旧版本为cn.clickwise.clickad.radiusClient, 新版本为

cn.clickwise.clickad.radiusReform

这里介绍最新版本的代码实现：

RemoteRadiusReformNoRecursive：通过tcp协议连接RadiusCenter，然后按照既定格式依次读取packet(包括消息头和消息体，先解析消息头，获得消息体的长度信息，然后读取消息体并解析),每隔一定时间，发送心跳信息。

task(RadiusCenter rc):读取过程的控制流程，包括连接connect(rc)，sendHeartbeat()，readPacket()的调用顺序和时间间隔。

connect(RadiusCenter rc):连接RadiusCenter

readPacket():读取一个packet,读取消息头、读取消息体。调用sockIn.read(Buffer.head)

读取消息头，调用parseBytes2Info()获得消息体的长度。调用

sockIn.read(Buffer.body,0,Buffer.packetbodylen-12)读取消息体。最后调

用receiveNoAnalysisCompletelyPacketBody()解析消息体。

parseBytes2Info():head的前四个字节表示packet的总长度。调用

BytesTransform.byteToIntv(Buffer.fbyte)将前四个字节转化成int类型。

receiveNoAnalysisCompletelyPacketBody()：解析消息体，取出包含user\_name、frameipaddress、status-type的字节，并将这些字节转换成十六进制字符串的形式(这三个字段的顺序不固定)。 加上时间戳，形成rawRecord，将rawRecord写入日志文件。并将rawRecord通过tcp协议发到远程机器。在远程机器上解析。

RemoteResolve:远程机器上的解析服务。启动server，接受客户端通过tcp协议接入。然后接收发过来的rawRecord，rawRecord添加到记录池(QueueRecordPond)中。

init():初始化函数，通过confFactory.getRSPort()获得服务端口，在该端口启动ServerSocket。每个接入的socket，都启动一个ServerThread。接收传入的rawRecord。

内部类ServerThread：sockIn.readInt()读取包的长度n(单位为char)。然后读取n个char，转换成字符串，加入到记录池(QueueRecordPond)中。

QueueRecordPond：使用ConcurrentLinkedQueue保存记录，启动多个线程，从队列中提取rawRecord，解析结果存入日志文件，并更新在线radius库，即当前ip对应的radiusId和 radiusId当前分配的ip。

void initForParse():初始化 redis配置，使用redis存储在线的radius数据。

void add2Pond(String record): 将record加入到ConcurrentLinkedQueue中

String pollFromPond():从ConcurrentLinkedQueue中取record。

void startConsume(int threadNum)：启动FieldResolve线程

FieldResolve:内部类，初始化各自线程写入的文件名， 以及redis配置信息。从ConcurrentLinkedQueu中poll record。并调用RadiusAnalysis的analysis(record)方法，解析rawRecord。将结果写入文件，并更

新在线radius库。

RadiusAnalysis： 提供rawRecord的analysis方法，同时也负责离线日志数据的解析。即读取输入流，解析并将结果写入输出流。

RecordLight analysis(String line):分析rawRecord，将其中的时间戳、radiusID、frameIPArress、statusType转换成标准格式。

分词、词性标注、关键词及服务,

此部分包括

cn.clickwise.clickad.sample、cn.clickwise.clickad.server、cn.clickwise.clickad.seg、cn.clickwise.tag、cn.clickwise.clickad.keyword等部分。

cn.clickwise.clickad.seg. Segmenter: 对 ansj分词的封装

void loadAnsjDic(File dict)：加载词典文件

String segAnsi(String text)：对一段文本进行分词，返回分词后的结果

cn.clickwise.clickad.tag.PosTagger:对stanford postagger的封装

PosTagger(String model)：加载词性标注模型model

tag(String text)：对一段文本进行词性标注，返回词性标注后的结果

cn.clickwise.clickad.keyword.KeyExtract：提取关键词

keyword\_extract(String text):从分词并加词性标注后的文本提取关键词

keyword\_extract\_noun(String text):从分词并加词性标注后的文本提取名词

keyword\_extract\_noun\_ngram(String text): 从分词并加词性标注后的文本提取名词以及它们的ngram

keyword\_extract\_noun\_ngram\_vv\_adj(String text): 从分词并加词性标注后的文本提取名词以及它们的ngram、动词、形容词

cn.clickwise.clickad.server：

NLPServer：启动服务，提供分词、词性标注、关键词提取的http接口

KeyHandler：处理关键词请求的逻辑

SegHandler：处理分词请求的逻辑

TagHandler：处理词性标注请求的逻辑

cn.clickwise.clickad.sample:

Sampler:从标记好的普通样本(例如格式为label\001text)生成词典文件、标记索引文件和模型的标准样本(svm的输入样本 格式为

labelIndex<separator>[<wordIndex:count>\*])。类别标记为普通的一层分类。

SampleMulLevel：从标记好的普通样本(例如格式为labels\001text)生成词典文件、标记索引文件和模型的标准样本(svm的输入样本 格式为

labelIndexs<separator>[<wordIndex:count>\*])。类别标记为多层分类。Labels和labelIndexs，包含不同层次类别名称或类别标记，不同层次之间用分隔符例如”\_”分隔开。

* 十大行业网站

该部分代码位于cn.clickwise.mapreduce.app中

cn.clickwise.mapreduce.app.appstore\_analysis：对app\_store的分析。

cn.clickwise.mapreduce.app.estate\_analysis：房产领域分析

cn.clickwise.mapreduce.app.finance\_analysis：金融领域分析

cn.clickwise.mapreduce.app.game\_analysis：游戏领域分析

cn.clickwise.mapreduce.app.travel\_analysis：旅游领域分析

cn.clickwise.mapreduce.app.video\_analysis：视频领域分析

具体参见行业网站部分的文档。

* Rpc库

Rpc库是在远程执行命令，并将结果传回客户端的库

EasyServer启动rpc服务，接收传入的command ，反序列化解析出command，在本地执行，编译执行command将结果返回给客户端

有不同类型的command，每个command都对应三个类：

\*\*\*Command,\*\*\*Client,\*\*\*Handler

例如

从远程拷贝文件对应：

FileCopyFromClient、FileCopyFromCommand、FileCopyFromHandler

拷贝文件到远程对应：

FileCopyToClient、FileCopyToCommand、FileCopyToHandler

读取远程的hive表：

HiveFetchTableClient、HiveFetchTableCommand、HiveFetchTableHandler

读取远程的hive表，只取key包含在输入文件里的记录：

HiveFetchByKeysClient、HiveFetchByKeysCommand、HiveFetchByKeysHandler

读取远程的hive表，只取key包含在输入文件里的记录，并进行统计：

HiveStatisticByKeysClient、HiveStatisticByKeysCommand、HiveStatisticByKeysHandler

* 工具库(tool)

工具库的代码位于cn.clickwise.lib下：

cn.clickwise.lib.bytes(byte数组的各种操作)

BytesTransformbyte 数组转换成各种类型的数据

例如byteToInt2

byteToIntv

bytes2str

cn.clickwise.lib.code(各种编码和加密方法)

MD5Code:使用md5算法加密字符串

Base64Code：使用base64算法编码或解码字符串

Unicode：unicode编码或解码字符串

UrlCode:urlcode编码或解码字符串

cn.clickwise.lib.file(文件操作的便利类)

FileReaderUtil(将文件转换成各种数据结构，例如hashMap、list等)

FileStatus(测试文件的各种状态)

PropertiesUtil(将文件转换成Properties对象)

cn.clickwise.lib.jar

GenConf(生成ant的配置文件build.xml的配置信息)

cn.clickwise.lib.linux

COMMAND(执行linux脚本)

cn.clickwise.lib.sort

SortStrArray(排序的便利类，对数组或list排序，返回排序结果)

cn.clickwise.lib.string(字符串的各种操作)

SSO(Simple Str Operation 简单的字符串处理类)

包括：

booleantnoe(String str)测试字符串是否为null或空,,若不为null且不为

空返回true

booleantioe(String str)测试字符串是否为null或空,,若为null或为空返回true

String implode(String[] seg\_arr, String separator)

合并字符串：将seg\_arr中的元素用separator分割，然后返回字符串

String implode(ArrayListarr\_list, String separator)

合并字符串：将arr\_list中的元素用separator分割，然后返回字符串

String beforeStr(String source, String str)

取str之前的字符串,不包括str。

String afterStr(String source, String str)

取str之后的字符串,不包括str

String truncAfterStr(String source, String str)

截断str之后的字符串，包括str

String truncBeforeStr(String source, String str)

截断str之前的字符串，包括str

String midstrs(String source, String str1, String str2)

返回str1,str2中间的字符串，不包括str1,str2(str1 第一次出现，str2 最后

一次出现)

String midfstrs(String source, String str1, String str2)

返回str1,str2中间的字符串，不包括str1,str2(str1 第一次出现，str2 第一

次出现)

String replaceLast(String source, String find, String replace)

将source最后出现的find 替换为replace

String ngram (String word)

获得一个单词的ngram，返回ArrayList<String>

RegexFind(根据正则表达式找到匹配的结果)

String[] find(String regex,Stringline,int n)

用正则表达式匹配line,返回结果数组。regex:匹配表达式, line: 被匹配的字符串, n:生成的字段数目

String findSingle(String regex,String line)

用正则表达式匹配line,返回结果(只有一个)。regex:匹配表达式, line: 被匹配的字符串

RegexDB:存储各种常用的正则表达式，例如匹配url、中文、urlcode编码的正则表达式

cn.clickwise.lib.time

TimeOpera:时间处理的工具类

包括字符串格式和long格式的时间的相互转换，获取当前时间、昨天、n天前的日期和时间的各种表示。

* Web抓取库(网页抓取的类库)

Fetcher: 抓取网页的各种内容，例如源代码、title、keywords、description等

QueueUrlPond：从queue中取url，进行抓取，结果输出到文件中

FieldCrawl：抓取一行某个字段的内容，并将内容加到改行的末尾