



SAS[®] Contextual Analysis In- Database Scoring 14.3 for Hadoop: User's Guide

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2017. *SAS® Contextual Analysis In-Database Scoring 14.3 for Hadoop: User's Guide*. Cary, NC: SAS Institute Inc.

SAS® Contextual Analysis In-Database Scoring 14.3 for Hadoop: User's Guide

Copyright © 2017, SAS Institute Inc., Cary, NC, USA

All Rights Reserved. Produced in the United States of America.

For a hard copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

September 2017

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

14.3-P1:ctxtindbug

Contents

SAS Contextual Analysis In-Database Scoring 14.3 for Hadoop: User's Guide v
What's New in SAS SAS Contextual Analysis In-Database Scoring 14.3 for Hadoop . . . vii

Chapter 1 • Introduction	1
About SAS Contextual Analysis In-Database Scoring for Hadoop	1
Chapter 2 • Using the Text Analytics Models	3
Accessing the Models	3
Using the Concept Model	3
Using the Category Model	8
Using the Sentiment Model	12
Recommended Reading	21

SAS Contextual Analysis In-Database Scoring 14.3 for Hadoop: User's Guide

Audience

This book is for users who want to work with SAS Contextual Analysis scoring models in Hadoop. It is assumed that you know how to use SAS Contextual Analysis.

Requirements

You must have the following products licensed:

- SAS Contextual Analysis
- SAS In-Database Code Accelerator for Hadoop

Before you can use the In-Database Scoring for Hadoop, you must follow the steps in Chapters 2 and 3 of *SAS Contextual Analysis In-Database Scoring for Hadoop: Administrator's Guide*.

What's New in SAS SAS Contextual Analysis In-Database Scoring 14.3 for Hadoop

New Sentiment DS2 Score Code

The sentiment analysis DS2 score code now includes information about product- and feature-level sentiment. (Previous releases included document-level sentiment only).

Chapter 1

Introduction

About SAS Contextual Analysis In-Database Scoring for Hadoop 1

About SAS Contextual Analysis In-Database Scoring for Hadoop

This product enables you to run scoring models generated in SAS Contextual Analysis inside a Hadoop environment. Sample code that you would run is provided here. Please see the appropriate sections of this book for your needs.

Item to Score	See This Section
Concepts	“Using the Concept Model” on page 3
Categories	“Using the Category Model” on page 8
Sentiment	“Using the Sentiment Model” on page 12

Chapter 2

Using the Text Analytics Models

Accessing the Models	3
Using the Concept Model	3
Using the Category Model	8
Using the Sentiment Model	12

Accessing the Models

Text analytics models include score code, which is generated within SAS Contextual Analysis. To generate the score code, see the section “Viewing and Downloading Code” in Chapter 2 of *SAS Contextual Analysis: User’s Guide*.

- To locate the binary files for concepts and categories, see “Locating the Rules Files (LI and MCO)” in *SAS Contextual Analysis: Administrator’s Guide*.
- To locate language-specific SAM (binary) files that are used in SAS Contextual Analysis, see the file `\tktg\sasmisc\nn-base.sam` (Windows) or `misc/tktg/nn-base.sam` (UNIX) under your SAS installation directory. Note that the first two characters (*nn*) of the .sam filename denote the licensed language. For example, in `en-base.sam`, the `en` denotes English.
- To locate sentiment (SAM) files that you created using SAS Sentiment Analysis Studio, see *SAS Sentiment Analysis Studio 12.2: User’s Guide*.

Using the Concept Model

The following code can be executed in SAS Contextual Analysis In-Database Scoring for Hadoop for concept extraction.

```

/*****
* Concept score code for Hadoop
*
* Copyright (C), 2016
* SAS Institute Inc., Cary, N.C. 27513, U.S.A. All rights reserved.
*****/

```

```

/*****
 * Set SAS environment variables that specify the location of the Hadoop Java
 * client API and configuration files. They are used to access Hadoop services.
 * The Java client API is provided by the Hadoop vendor in the form of JAR files.
 *
 * NOTE: The SAS Contextual Analysis In-Database Scoring for Hadoop
 *       Administrator's Guide describes how to collect Hadoop Java client API jars
 *       and configuration files.
 *****/
options set=SAS_HADOOP_JAR_PATH="C:\path\to\Hadoop\jars";
options set=SAS_HADOOP_CONFIG_PATH="C:\path\to\Hadoop\conf";

/*****
 * Execute a LIBNAME statement to assign a library reference to associate with
 * a Hadoop HDFS or HIVE server.
 *
 * Please contact your IT administrator for the HIVE server name.
 *****/
libname gridlib hadoop server="hivenode.com" user=xxxxx password=xxxxxxx;

/*****
 * Execute a LIBNAME statement to assign a library reference to
 * the location of the local SAS data set to be scored.
 *
 * If all the data sets used for scoring are already in HDFS, the library might
 * not be needed.
 *****/
libname home "C:\path\to\local\dataset";

/*****
 * Copy the data set to HDFS. Unique observation IDs are required and are
 * created as the field _document_id.
 *
 * If all the data sets used for scoring are already in HDFS, this step might
 * not be needed.
 *****/
data gridlib.input_dataset;
  set home.input_dataset;
  _document_id = _N_;
run;

/*****
 * Set input/output macro variables.
 *
 * input_ds:          Name of input data set in Hadoop, including the HDFS libref
 * document_id:       Name of the unique document ID column in the input data set
 * document_column:   Name of the column to process in the input data set
 * liti_binary_path:  Path to the LITI binary
 * output_ds:         Name of the output data set, including the HDFS libref
 *
 * NOTE: The SAS Contextual Analysis In-Database Scoring for Hadoop
 *       Administrator's Guide describes how to deploy Text Analytics models
 *       into clusters.
 *****/
%let input_ds = gridlib.input_dataset;
%let document_id = _document_id;

```

```

%let document_column = text_to_process;
%let liti_binary_path = '/path/to/liti/binary.li';
%let output_ds = gridlib.concept_out;

/*****
 * Delete the output before starting (Optional)
 *****/
proc delete data=&output_ds; run;

/*****
 * Scores concepts in Hadoop
 *****/
proc ds2 ds2accel=yes xcode=warning;

    /* These packages are part of the Text Analytics add-on and are installed */
    /* in the EP */
    require package tkcat; run;
    require package tktxtanio; run;

    /* The output of the thread program is the input of the data program */
    THREAD workerth / overwrite=YES;

    drop _apply_settings _document _liti_binary _trans;

    dcl package tkcat cat();
    dcl package tktxtanio txtanio();

    dcl binary(8)    _apply_settings;
    dcl binary(8)    _document;
    dcl binary(8)    _liti_binary;
    dcl binary(8)    _trans;

    dcl double      _status;
    dcl double      _num_matches;
    dcl double      _i;
    dcl double      _document_id;
    dcl varchar(1024) _name;
    dcl varchar(1024) _full_path;
    dcl double      _start_offset;
    dcl double      _end_offset;
    dcl varchar(1024) _term;
    dcl varchar(1024) _canonical_form;

    retain          _apply_settings;
    retain          _liti_binary;
    retain          _trans;

/*****
 * Initialization step. Only runs once when starting.
 *****/
method init();
    _apply_settings = cat.new_apply_settings();
    _liti_binary = txtanio.new_on_content_server(&liti_binary_path);

    _status = cat.set_apply_model(_apply_settings, _liti_binary);
    if _status NE 0 then put 'ERROR: set_apply_model fails';

```

```

/* Match types are 0=ALL, 1=LONGEST or 2=BEST */
_status = cat.set_match_type(_apply_settings, 0);
if _status NE 0 then put 'ERROR: set_match_type fails';

_status = cat.initialize_concepts(_apply_settings);
if _status NE 0 then put 'ERROR: initialize_concepts fails';

_trans = cat.new_transaction();
end;

/*****
* Run step. The method runs per row of input.
*****/
method run();
  set &input_ds(keep=(&document_column &document_id));

  /* Only process if document observation is not empty*/
  if &document_column NE ' ' then do;

    /* Initialize the document with the column data */
    _document = txtanio.new_document_from_string(&document_column);

    /* Set the document on the transaction so we're ready to process */
    _status = cat.set_document(_trans, _document);
    if _status NE 0 then put 'ERROR: set_document fails on obs:' &document_id;

    /* Apply the binary to the document */
    _status = cat.apply_concepts(_apply_settings, _trans);
    if _status NE 0 then put 'ERROR: apply_concepts fails on obs:' &document_id;

    /* Look for the concept matches */
    _num_matches = cat.get_number_of_concepts(_trans);
    _i = 0;
    do while (_i LT _num_matches);
      _name = cat.get_concept_name(_trans, _i);
      _full_path = cat.get_full_path_from_name(_trans, _name);
      _start_offset = cat.get_concept_start_offset(_trans, _i);
      _end_offset = cat.get_concept_end_offset(_trans, _i);
      _term = cat.get_concept(_trans, _i);
      _canonical_form = cat.get_concept_canonical_form(_trans, _i);

      output;

      _i = _i + 1;
    end;

    /* Now look for fact matches */
    _num_matches = cat.get_number_of_facts(_trans);
    _i = 0;
    _canonical_form = '';
    do while (_i LT _num_matches);
      _name = cat.get_fact_name(_trans, _i);
      _full_path = cat.get_full_path_from_name(_trans, _name);
      _start_offset = cat.get_fact_start_offset(_trans, _i);
      _end_offset = cat.get_fact_end_offset(_trans, _i);

```

```

        _term = cat.get_fact(_trans, _i);

        output;

        _i = _i + 1;
    end;
    _i = 0;

    /* Clean up resources */
    cat.clean_transaction(_trans);
    txtanio.free_object(_document);
end;
end;

/*****
 * Termination step that runs only once at the end.
 *****/
method term();
    /* clean up resources */
    cat.free_transaction(_trans);
    cat.free_apply_settings(_apply_settings);
    txtanio.free_object(_liti_binary);
end;
endthread;
run;

/*****
 * Collect output data
 *****/
data &output_ds(
    keep=(
        &document_id
        _name
        _full_path
        _start_offset
        _end_offset
        _term
        _canonical_form
    )
    overwrite=yes
);
dcl THREAD workerth THRD;

method run();
    set from THRD;
end;

enddata;
run; quit;

```

Using the Category Model

The following code can be executed in SAS Contextual Analysis In-Database Scoring for Hadoop for context categorization.

```

/*****
 * Concept score code for Hadoop
 *
 * Copyright (C), 2016
 * SAS Institute Inc., Cary, N.C. 27513, U.S.A. All rights reserved.
 *****/

/*****
 * Set SAS environment variables that specify the location of the Hadoop Java
 * client API and configuration files. They are used to access Hadoop services.
 * The Java client API is provided by the Hadoop vendor in the form of JAR files.
 *
 * NOTE: The SAS Contextual Analysis In-Database Scoring for Hadoop
 *       Administrator's Guide denotes how to collect Hadoop Java client API jars
 *       and configuration files.
 *****/
options set=SAS_HADOOP_JAR_PATH="C:\path\to\Hadoop\jars";
options set=SAS_HADOOP_CONFIG_PATH="C:\path\to\Hadoop\conf";

/*****
 * Execute a LIBNAME statement to assign a library reference to associate with
 * a Hadoop HDFS or HIVE server.
 *
 * Please contact your IT administrator for the HIVE server name.
 *****/
libname gridlib hadoop server="hivenode.com" user=xxxxx password=xxxxxxx;

/*****
 * Execute a LIBNAME statement to assign a library reference to
 * the location of the local SAS data set to be scored.
 *
 * If all the data sets used for scoring are already in HDFS, the library may
 * not be needed.
 *****/
libname home "C:\path\to\local\dataset";

/*****
 * Copy the data set to HDFS. Unique observation IDs are required and are
 * created as the field _document_id.
 *
 * If all the data sets used for scoring are already in HDFS, this step might
 * not be needed.
 *****/
data gridlib.input_dataset;
    set home.input_dataset;
    _document_id = _N_;
run;

```



```

/*****
 * Set input/output macro variables.
 *
 * input_ds:          Name of input data set in Hadoop, including the HDFS libref
 * document_id:       Name of the unique document ID column in the input data set
 * document_column:   Name of the column to process in the input data set
 * liti_binary_path:  Path to the LITI binary
 * output_ds:         Name of the output data set, including the HDFS libref
 *
 * NOTE: The SAS Contextual Analysis In-Database Scoring for
 *       Hadoop Administrator's Guide describes how to deploy Text Analytics models
 *       into clusters.
 *****/
%let input_ds = gridlib.input_dataset;
%let document_id = _document_id;
%let document_column = text_to_process;
%let mco_binary_path = '/path/to/mco/binary.mco';
%let output_ds = gridlib.category_out;

/*****
 * Delete the output before starting (Optional)
 *****/
proc delete data=&output_ds; run;

/*****
 * Score categories in Hadoop
 *****/
proc ds2 ds2accel=yes xcode=warning;

    /* These packages are part of the Text Analytics add-on and are installed */
    /* in the EP                                                                */
    require package tkcat; run;
    require package tktxtanio; run;

    /* The output of the thread program is the input of the data program      */
    THREAD workerth / overwrite=YES;

    drop _apply_settings _document _mco_binary _trans;

    dcl package tkcat cat();
    dcl package tktxtanio txtanio();

    dcl binary(8)    _apply_settings;
    dcl binary(8)    _document;
    dcl binary(8)    _mco_binary;
    dcl binary(8)    _trans;

    dcl double      _status;
    dcl double      _num_matches;
    dcl double      _num_terms;
    dcl double      _i;
    dcl double      _j;
    dcl double      _document_id;
    dcl varchar(1024) _name;
    dcl varchar(1024) _full_path;
    dcl double      _start_offset;

```

```

        dcl double      _end_offset;
        dcl varchar(1024) _term;

        retain          _apply_settings;
        retain          _mco_binary;
        retain          _trans;

/*****
 * Initialization step. Runs only once when starting.
 *****/
method init();
    _apply_settings = cat.new_apply_settings();
    _mco_binary = txtanio.new_on_content_server(&mco_binary_path);

    cat.set_categories_model(_apply_settings, _mco_binary);
    cat.set_return_match_positions_for_categories(_apply_settings, 1);
    cat.set_relevancy_type(_apply_settings, 1);

    _status = cat.initialize_categories(_apply_settings);
    if _status NE 0 then put 'ERROR: initialize_categories fails';

    _trans = cat.new_transaction();
end;

/*****
 * Run step. The method runs per row of input.
 *****/
method run();
    set &input_ds(keep=(&document_column &document_id));

    /* Only process if document observation is not empty*/
    if &document_column NE ' ' then do;

        /* Initialize the document with the column data */
        _document = txtanio.new_document_from_string(&document_column);

        /* Set the document on the transaction so we're ready to process */
        _status = cat.set_document(_trans, _document);
        if _status NE 0 then put 'ERROR: set_document fails on obs:' &document_id;

        /* Apply the binary to the document */
        _status = cat.apply_categories(_apply_settings, _trans);
        if _status NE 0 then put 'ERROR: apply_categories fails on obs:' &document_id;

        /* Get the number of categories matches */
        _num_matches = cat.get_nb_matched_categories(_trans);
        _i = 0;
        do while (_i LT _num_matches);
            /* we used the name for both name & full_path for better alignment */
            /* with LITI code */
            _name = cat.get_category_name(_trans, _i);
            _full_path = _name;

            /* Get the number of term matches for current matched category */
            _num_terms = cat.get_nb_matched_terms_category(_trans, _i);
            _j = 0;

```

```

do while (_j LT _num_terms);
    _start_offset = cat.get_matched_term_start_offset_category(_trans, _i, _j);
    _end_offset = cat.get_matched_term_end_offset_category(_trans, _i, _j);
    _term = cat.get_matched_term_category(_trans, _i, _j);

    output;

    _j = _j + 1;
end;

_i = _i + 1;
end;

/* Clean up resources */
cat.clean_transaction(_trans);
txtanio.free_object(_document);
end;
end;

/*****
* Termination step. Runs only once at end.
*****/
method term();
    /* Clean up resources */
    cat.free_transaction(_trans);
    cat.free_apply_settings(_apply_settings);
    txtanio.free_object(_mco_binary);
end;

endthread;
run;

/*****
* Collect output data
*****/
data &output_ds(
    keep=(
        &document_id
        _name
        _full_path
        _start_offset
        _end_offset
        _term
    )
    overwrite=yes
);
dcl THREAD workerth THRD;

method run();
    set from THRD;
end;

enddata;
run; quit;

```

Using the Sentiment Model

The following code can be executed in SAS Contextual Analysis In-Database Scoring for Hadoop for analyzing sentiment.

```

/*****
 * Sentiment score code for Hadoop
 *
 * Copyright (C), 2017
 * SAS Institute Inc., Cary, N.C. 27513, U.S.A. All rights reserved.
 *****/

/*****
 * Set SAS environment variables that specify the location of the Hadoop Java
 * client API and configuration files. They are used to access Hadoop services.
 * The Java client API is provided by Hadoop vendor in the form of JAR files.
 *
 * NOTE: The Admin Guide of SAS Contextual Analysis In-Database Scoring for
 *       Hadoop denotes how to collect Hadoop Java client API jars and
 *       configuration files.
 *****/
options set=SAS_HADOOP_JAR_PATH="C:\path\to\Hadoop\jars";
options set=SAS_HADOOP_CONFIG_PATH="C:\path\to\Hadoop\conf";

/*****
 * Execute a LIBNAME statement to assign a library reference to associate with
 * a Hadoop HDFS or HIVE server.
 *
 * Please contact your IT administrator for HIVE server name.
 *****/
libname dblib hadoop server="hivenode.com" user=xxxxx password=xxxxxxx;

/*****
 * The LIBNAME statement to reference to local SAS dataset. The library is
 * used to copy local SAS dataset to HDFS.
 *
 * If all the datasets used for scoring are already in HDFS, the library may
 * not be needed.
 *****/
libname home "C:\path\to\local\dataset";

/*****
 * Copy dataset to HDFS. It also sets unique observation ids.
 *
 * If all the datasets used for scoring are already in HDFS, this step may
 * not be needed.
 *****/
data dblib.input_dataset;
    set home.input_dataset;
    _document_id = _N_;
run;

```

```

/*****
 * Set input/output macro variables.
 *
 * input_ds:          Name of input dataset in Hadoop
 * document_column:  Name of the column to process in the input dataset
 * sam_binary_path:  Path to the SAM binary
 * output_ds:        Output document level sentiment and matched string dataset
 * output_ds_feat:   Output feature level sentiment dataset
 *
 * NOTE: The Admin Guide of SAS Contextual Analysis In-Database Scoring for
 *       Hadoop describes how to deploy Text Analytics models into clusters.
 *****/
%let input_ds = dblink.input_dataset;
%let document_id = _document_id;
%let document_column = text_to_process;
%let sam_binary_path = '/path/to/sam/binary.sam';
%let output_ds = dblink.sentiment_out;
%let output_ds_feat = dblink.sentiment_feat_out;

/*****
 * Delete output before start (OPTIONAL)
 *****/
proc delete data=&output_ds run;
proc delete data=&output_ds_feat run;

/*****
 * Scores document level sentiment and matched string information in Hadoop
 *****/
proc ds2 ds2accel=yes xcode=warning;

    /* These packages are part of the Text Analytics add-on and are installed */
    /* in the EP                                                                */
    require package tksent; run;
    require package tktxtanio; run;
    require package tkling; run;

    /* The output of the thread program is the input of the data program      */
    THREAD workerth / overwrite=YES;

        drop _apply_settings _document _sam _trans;

        dcl package tksent sent();
        dcl package tktxtanio txtanio();
        dcl package tkling ling();

        dcl binary(8) _apply_settings;
        dcl binary(8) _document;
        dcl binary(8) _sam;
        dcl binary(8) _trans;
        dcl char(300) _binfile;
        dcl char(100) _match_rule_type;
        dcl double _match_count;
        dcl double _match_index;
        dcl double _rule_type;
        dcl double _status;

```

```

dcl char(10)  _sentiment;
dcl double   _sentiment_probability;
dcl double   _char_start;
dcl double   _char_end;
dcl char(100) _match_string;
dcl char(20)  _match_sentiment;
dcl char(100) _product_name;
dcl char(100) _feature_path;

retain      _apply_settings;
retain      _sam_binary;
retain      _trans;

/*****
 * Initialization step. Only runs once when starting.
 *****/
method init();
    _apply_settings = sent.new_apply_settings();

    _status = sent.set_threshold(_apply_settings, 0.5);
    if _status NE 0 then put 'ERROR: set_threshold fails';

    _sam = txtanio.new_on_content_server(&sam_binary_path);

    _status = sent.set_model(_apply_settings, _sam);
    if _status NE 0 then put 'ERROR: set_model fails';

    _trans = sent.new_transaction();

    _status = sent.initialize_sentiment(_apply_settings, _trans);
    if _status NE 0 then put 'ERROR: initialize_sentiment fails';
end;

/*****
 * Run step. The method excutes or runs per row of input.
 *****/
method run();
    set &input_ds(keep=(&document_column &document_id));

    /* Only process if document observation is not empty*/
    if &document_column NE ' ' then do;

        /* Initialize the document with the column data */
        _document = txtanio.new_document_from_string(&document_column);

        /* Set the document on the transaction so we're ready to process */
        _status = sent.set_document(_trans, _document);
        if _status NE 0 then put 'ERROR: set_document fails on document:' &document_id

        /* Apply the binary to the document */
        _status = sent.apply_sentiment(_apply_settings, _trans);
        if _status NE 0 then put 'ERROR: apply_sentiment fails on document:' &document_id

        /* Get current document sentiment polarity and probability */
        _sentiment = sent.get_sentiment(_trans);
        _sentiment_probability = sent.get_probability(_trans);

```

```

/* get matched string count within the text */
_rule_type = 0;

do;
  do while (_rule_type LE 5);

    _match_count = sent.get_rule_match_count(_trans, _rule_type);
    _match_index = 0;

    /* if there are matched strings */
    if _match_count GT 0 then
      do;
        /* get matched string sentiment */
        do while (_match_index LT _match_count);
          _char_start = sent.get_char_start_offset
            (_trans, _rule_type, _match_index);
          _char_end = sent.get_char_end_offset
            (_trans, _rule_type, _match_index);
          _match_string = sent.get_match_string
            (_trans, _rule_type, _match_index);
          _match_sentiment = sent.get_match_string_sentiment
            (_trans, _rule_type, _match_index);
          _product_name = sent.get_match_string_product_name
            (_trans, _rule_type, _match_index);
          _feature_path = sent.get_match_string_feature_path
            (_trans, _rule_type, _match_index);

          output;
          _match_index = _match_index + 1;
        end;
      end;

    _rule_type = _rule_type + 1;
  end;
end;

/* Clean up resources */
sent.clean_sentiment(_trans);
txtanio.free_object(_document);
end;

/*****
* Termination step. Only runs once at end.
*****/
method term();
  /* clean up resources */
  sent.free_transaction(_trans);
  sent.free_apply_settings(_apply_settings);
  txtanio.free_object(_sam);
end;

endthread;
run;

```

```

/*****
 * Collect output data
 *****/
data &output_ds(
    keep=(
        &document_id
        _sentiment
        _sentiment_probability
        _char_start
        _char_end
        _match_string
        _match_sentiment
        _product_name
        _feature_path
    )
    overwrite=yes
);
dcl THREAD workerth THRD;

method run();
    set from THRD;
end;
enddata;
run; quit;

/*****
 * Scores feature level sentiment in Hadoop
 *****/
proc ds2 ds2accel=yes xcode=warning;

    /* These packages are part of the Text Analytics add-on and are installed */
    /* in the EP */
    require package tksent; run;
    require package tktxtanio; run;
    require package tkling; run;

    /* The output of the thread program is the input of the data program */
    THREAD workerth / overwrite=YES;

    drop _apply_settings _document _sam _trans;

    dcl package tksent sent();
    dcl package tktxtanio txtanio();
    dcl package tkling ling();

    dcl binary(8) _apply_settings;
    dcl binary(8) _document;
    dcl binary(8) _sam;
    dcl binary(8) _trans;
    dcl char(300) _binfile;
    dcl double _product_count;
    dcl double _product_index;
    dcl double _feature_count;
    dcl double _feature_index;
    dcl double _status;

```



```

dcl char(100)  _product_name;
dcl char(20)   _product_sentiment;
dcl double    _product_probability;
dcl char(100) _feature_path;
dcl char(20)  _feature_sentiment;
dcl double    _feature_probability;

retain        _apply_settings;
retain        _sam_binary;
retain        _trans;

/*****
 * Initialization step. Only runs once when starting.
 *****/
method init();
    _apply_settings = sent.new_apply_settings();

    _status = sent.set_threshold(_apply_settings, 0.5);
    if _status NE 0 then put 'ERROR: set_threshold fails';

    _sam = txtanio.new_on_content_server(&sam_binary_path);

    _status = sent.set_model(_apply_settings, _sam);
    if _status NE 0 then put 'ERROR: set_model fails';

    _trans = sent.new_transaction();

    _status = sent.initialize_sentiment(_apply_settings, _trans);
    if _status NE 0 then put 'ERROR: initialize_sentiment fails';
end;

/*****
 * Run step. The method excutes or runs per row of input.
 *****/
method run();
    set &input_ds(keep=(&document_column &document_id));

    /* Only process if document observation is not empty*/
    if &document_column NE ' ' then do;

        /* Initialize the document with the column data */
        _document = txtanio.new_document_from_string(&document_column);

        /* Set the document on the transaction so we're ready to process */
        _status = sent.set_document(_trans, _document);
        if _status NE 0 then put 'ERROR: set_document fails on document:' &document_id

        /* Apply the binary to the document */
        _status = sent.apply_sentiment(_apply_settings, _trans);
        if _status NE 0 then put 'ERROR: apply_sentiment fails on document:' &document_id

    do;
        /* get product level sentiment */
        _product_count = sent.get_product_count(_trans);
        _product_index = 0;

```

```

if _product_count GT 0 then
do;
do while (_product_index LT _product_count);
_product_name = sent.get_product_name
(_trans, _product_index);
_product_sentiment = sent.get_product_sentiment
(_trans, _product_index);
_product_probability = sent.get_product_probability
(_trans, _product_index);

/* get feature level sentiment */
_feature_count = sent.get_feature_count(_trans, _product_index);
_feature_index = 0;

if _feature_count GT 0 then
do;
do while (_feature_index LT _feature_count);
_feature_path = sent.get_feature_path
(_trans, _product_index, _feature_index);
_feature_sentiment = sent.get_feature_sentiment
(_trans, _product_index, _feature_index);
_feature_probability = sent.get_feature_probability
(_trans, _product_index, _feature_index);

output;
_feature_index = _feature_index + 1;
end;
end;

if _feature_count EQ 0 then
do;
output;
end;

_product_index = _product_index + 1;
end;
end;
end;

/* Clean up resources */
sent.clean_sentiment(_trans);
txtanio.free_object(_document);
end;
end;

/*****
* Termination step. Only runs once at end.
*****/
method term();
/* clean up resources */
sent.free_transaction(_trans);
sent.free_apply_settings(_apply_settings);
txtanio.free_object(_sam);
end;

endthread;

```

```
run;

/*****
* Collect output data
*****/
data &output_ds_feat(
  keep=(
    &document_id
    _product_name
    _product_sentiment
    _product_probability
    _feature_path
    _feature_sentiment
    _feature_probability
  )
  overwrite=yes
);
dcl THREAD workerth THRD;

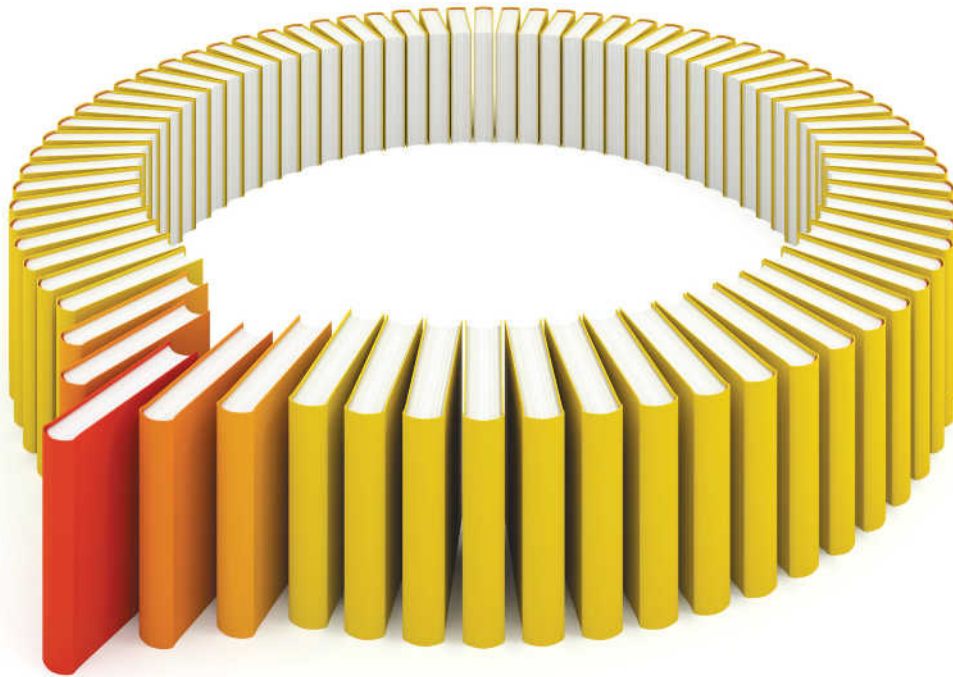
method run();
  set from THRD;
end;
enddata;
run; quit;
```


Recommended Reading

- *SAS Contextual Analysis: Administrator's Guide*
- *SAS Contextual Analysis: User's Guide*
- *SAS Contextual Analysis In-Database Scoring for Hadoop: User's Guide*
- *SAS In-Database Products: Administrator's Guide*
- *SAS In-Database Products: User's Guide*
- SAS Global Forum Paper, "Exploring SAS Embedded Process Technologies on Hadoop"

For a complete list of SAS publications, go to sas.com/store/books. If you have questions about which titles you need, please contact a SAS Representative:

SAS Books
SAS Campus Drive
Cary, NC 27513-2414
Phone: 1-800-727-0025
Fax: 1-919-677-4444
Email: sasbook@sas.com
Web address: sas.com/store/books



Gain Greater Insight into Your SAS[®] Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.

 support.sas.com/bookstore
for additional books and resources.


THE POWER TO KNOW.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies. © 2013 SAS Institute Inc. All rights reserved. S107969US.0613

