**Model Data Ingestion Pipelines with SAP Data Hub** DAT361

Exercises / Solutions  
Richard LeBlanc / SAP Labs LLC

Christian Krause / SAP SE

Christian Sengstock / SAP SE

Eric Westenberger / SAP SE

Shreepathi Kukkila / SAP Labs India

Mohana Andal / SAP Labs India

Table of ContentS

[Exercise overview 3](#_Toc525717113)

[Exercise 1 – Extract product reviews and perform sentiment analysis on them 4](#_Toc525717114)

[Exercise2 – COMBINE sentiment analysis data with product data 22](#_Toc525717115)

[Exercise3 – chart the results 41](#_Toc525717116)

# 

# Before You Start

## Browser

Please use supported browser(s) for your exercises. Chrome is preferred for this hands-on session.

## System Information

**SAP Data Hub Launchpad**

|  |  |
| --- | --- |
| URL (Group 01-05) | [https://ip-18-185-233-215.sapdatahub.com:9086](https://ip-18-185-233-215.sapdatahub.com:9086/) |
| URL (Group 06-10) | <https://ip-18-185-232-87.sapdatahub.com:9086> |
| URL (Group 11-15) | <https://ip-52-29-139-1.sapdatahub.com:9086> |
| URL (Group 16-20) | <https://ip-35-156-49-82.sapdatahub.com:9086> |
| URL (Group 21-25) | <https://ip-18-195-193-93.sapdatahub.com:9086> |
| URL (Group 26-30) | <https://ip-18-194-22-13.sapdatahub.com:9086> |
| Additional | <https://ip-18-195-61-115.sapdatahub.com:9086> |
| Additional | <https://ip-18-197-250-138.sapdatahub.com:9086> |
|  |  |
| Tenant name | default |
| Username | TAxx  Where xx is your group number |
| Password | Welcome01 |

**Record your**

# Group ID

# Exercise overview

In this exercise, you will be working with two data sources.

1. Product reviews that come from an online shop. This is usually referred to as big data.
2. Product data coming from an ERP system. This is usually referred to as enterprise data.

The goal of this exercise is to combine big data with enterprise data to help make better products to increase sales.

# Exercise 1 – Extract product reviews and perform sentiment analysis on them

In this first exercise, you will build a pipeline that extracts product reviews from HDFS, performs sentiment analysis on the reviews using Python and load the results in an SAP Vora table.

| Explanation | Screenshot |
| --- | --- |
| 1. Open Chrome and go the SAP Data Hub URL provided by the instructor. 2. Set the Tenant name to default 3. Set the Username to taXX where XX is your assigned group number. 4. Set Password to Welcome01 |  |
| 1. Click on the Modeler tile. |  |
|
| 1. Click on the + icon to create a new graph. |  |
| 1. Add the Read File operator to the graph by clicking it and dragging it onto the graph editor. |  |
| 1. Click on the Open Configuration icon. |  |
| 1. Set the Service to hdfs. |  |
| 1. Open the Connection Editor. |  |
| 1. Set the Configuration Type to Configuration Manager. |  |
| 1. Set the Connection ID to HDFS and click Save. |  |
| 1. Set the Path to /DAT361/Product\_Reviews. |  |
| 1. Set Recursive to true. 2. Set the Pattern to .\*txt.   Please note that this is a regular expression that will find all files ending with txt in the provided path. It is not the typical wildcard search used in Windows which would be \*.txt instead.   1. Set Only Read on Change to true. |  |
| 1. Add the Wiretap operator to the graph. |  |
| 1. Connect the outFileName output port of the Read File operator to the Wiretap operator. |  |
| 1. Click on the Save icon in the toolbar. |  |
| 1. Set the name to dx.taXX.sentiment\_analysis where xx is your assigned group number. |  |
| 1. Click on the Run icon and execute the pipeline. |  |
| 1. Make sure your graph is running before going to the next step. |  |
| 1. Right click on the Wiretap operator. 2. Click on Open UI |  |
| 1. You should see the file names of all the files being read. |  |
| 1. Stop the execution of your graph. |  |
| 1. Add the Python2Operator operator to the graph. |  |
| 1. Click on the Add Port icon. |  |
| To pass the contents of the HDFS files from the Read File operator to the Python2Operator, you must add an input port to the Python2Operator.   1. Set the Name to input. 2. Set the Type to message 3. Click on the Input radio button. 4. Click OK |  |
| 1. Connect the outFile port of the ReadFile operator to the input port of the Python2Operator operator.   Once the Read File operator has read all the files, it will notify the Python2Operator that there are no more files. |  |
| 1. Click on the Python2Operatory to see the icons. 2. Click on the Script icon. |  |
| 1. Delete the existing content in the Python script editor 2. Copy the contents of the SentimentAnalysisOnProductReviews.py file and paste them in the Python script editor.   The first part of the script contains code that would be written by a data scientist. It parses the contents of the HDFS files and uses the textblob library to extract the sentiments being expressed in the text (positive, negative, neutral).  The second part of the code is specific to Data Hub as it takes the input from the upstream operator, calls functions in the script for processing and passes the output to the downstream operator.  It also listens for a notification from the upstream Read File operator to indicate all files have been read and notifies the downstream operator. |  |
| 1. Close the script window and go back to the graph. |  |
| To pass the contents of the Python2Operator to the next operator, we must add an output port to the Python2Operator.   1. Click on the Add Port icon. |  |
| 1. Set the Name to output. 2. Set the Type to message. 3. Click on the Output radio button. 4. Click OK. |  |
| 1. Right click on the Python2Operator and click on Group.   By adding the Python2Operator in a group, we can further define its runtime environment. |  |
| The script in the Python2Operator is using the textblob Python library so we need to make sure that it’s available in the runtime environment.   1. Click on Group to see the icons. 2. Click on the Configuration icon. |  |
| 1. Add a new tag. |  |
| 1. Set the tag property to textblob 2. Set the version to 0.12.0. |  |
| 1. Connect the output port of the Python2Operator operator to the in port of the Vora Avro Ingestor operator to store the results of the Python processing in SAP Vora. |  |
| 1. Click on the Vora Avro Ingestor operator to see the icons. 2. Click on the Configuration icon. |  |
| The Vora Avro Ingestor can ingest data in Avro, JSON or CSV format. We’ll be using JSON because that is what the Python script is producing.   1. Set the format to json. |  |
| The defaultAvroSchema property is a JSON string used to define the structure of the table where the data will be stored in SAP Vora. If the table doesn’t exist, the operator will create it.     1. Copy the contents of the AvroSchema.txt file and paste them in the defaultAvroSchema property. |  |
| 1. Set the connectionType to connection. |  |
| 1. Open the Connection editor. |  |
| 1. Set the Configuration Type to Configuration Manager. |  |
| 1. Set the Connection ID to VORA. 2. Click on Save. |  |
| 1. Set the databaseSchema to “default\taXX”, where XX is your assigned group number. 2. Set the ingestionMode to UPSERT. |  |
| 1. Add the Message Stop Event to the graph. |  |
| 1. Connect the Vora Avro Ingestor operator to the Message Stop Event operator.   This operator is a Python based operator and it’s listening for a notification from the Vora Avro Ingestor to indicate that all files were read and processed. |  |
| 1. Add the Graph Terminator operator to the graph. |  |
| 1. Connect the Message Stop Event operator to the Graph Terminator operator. This will ensure the graph terminates correctly when the last record has been processed. |  |
| 1. Click on the Save icon to save the latest changes. |  |
| 1. Click on the Run icon and execute the pipeline. |  |
| 1. Wait until the pipeline has executed successfully. Now it’s time to see the results. |  |
| 1. From the SAP Data Hub launchpad, click on the Vora Tools tile. |  |
| 1. Double click on the default\taXX schema where XX is your assigned group number. |  |
| 1. Right click on PRODUCT\_REVIEWS and click on Data Preview. |  |
| 1. Congratulations! You’ve successfully completed the exercise.   To recap, you created a pipeline that reads product review files from HDFS, uses Python to extract sentiments from those product reviews and load them into SAP Vora. |  |

# Exercise2 – COMBINE sentiment analysis data with product data

In this exercise, you will build a pipeline that combines sentiment analysis data that you just loaded into SAP Vora in the previous exercise, with product data such as names, descriptions, categories and price from an HDFS file. Then the results will be stored in SAP Vora.

| Explanation | Screenshot |
| --- | --- |
| 1. Click on the + icon to create a new graph. |  |
| 1. Add the Workflow Trigger operator to the graph editor.   This operator is used to start a graph. Graphs having this operator can only contain operators from the Data Workflows section.  These types of graphs can be considered as orchestrating data processes rather than performing the data processing. |  |
|
|
|
| 1. Add the Pipeline operator to the graph. |  |
| 1. Connect the Workflow Trigger operator to the Pipeline operator. |  |
| 1. Click on the Pipeline operator. 2. Click on the Configuration icon. |  |
| 1. Set the Graph Name to the graph you created in the previous exercise. |  |
| 1. Add the Data Transform operator to the graph. |  |
| 1. Connect the Pipeline operator to the Data Transform operator. 2. Double click on the Data Transform operator. |  |
| 1. Add the Data Source operator to the Data Transform editor. |  |
| 1. Double click on the DataSource1 operator. |  |
| 1. Click on Browse. |  |
| 1. Select the HDFS connection and click OK. |  |
| 1. Click on Browse. |  |
| 1. Double click the DAT361 folder. |  |
| 1. Select the product.csv file and click OK. |  |
| 1. Click on the Back icon to go back to the Data Transform editor. |  |
| 1. Add the Data Source operator to the Data Transform editor. |  |
| 1. Double click on the DataSource2 operator. |  |
| 1. Click on Browse. |  |
| 1. Select VORA and click OK. |  |
| 1. Click Browse. |  |
| 1. Double click on the default\taXX schema where XX is your assigned group number. |  |
| 1. Select the PRODUCT\_REVIEWS table and click OK. |  |
| 1. Click on the Back icon to go back to the Data Transform operator editor. |  |
| 1. Add the Join transform to the Data Transform editor. |  |
| 1. Connect DataSource1 and DataSource2 to Join1. 2. Double click on the Join1 transform. |  |
| 1. Click on the ID column of Join1\_Input1 and drag it on the ID column of Join1\_Input2 to define the join condition between the two table. |  |
| 1. Click on Columns |  |
| 1. Drag the following columns from the two input sources into the target.   ID (Input 1)  MD5 (Input 2)  NAME (Input 1)  DESCRIPTION (Input 1)  CATEGORY (Input 1)  PRICE (Input 1)  TEXT (Input 2)  LENGTH (Input 2)  POLARITY (Input 2)  SUBJECTIVITY (Input 2) |  |
| 1. Click on the Back icon to go back to the Data Transform editor. |  |
| 1. Add the Data Target transform to the Data Transform editor. |  |
| 1. Connect the Join1 transform to the DataTarget1 transform. 2. Double click on DataTarget1. |  |
| 1. Click on Browse. |  |
| 1. Select VORA. |  |
| 1. Click on Browse. |  |
| 1. Double click on default\taXX where XX is your assigned group number. |  |
| 1. Select the Products table and click OK. |  |
| 1. Click OK. |  |
| 1. Click on the auto-map by name icon.   This will map the input columns coming from the Join operator to the columns of the target table in SAP Vora that have the exact same name. |  |
| Because the LENGTH, POLARITY and SUBJECTIVITY columns exist in the source but not the target, they were added to the target. Since the target already has TEXT\_LENGTH, TEXT\_POLARITY and TEXT\_SUBJECTIVITY, we don’t want these new columns in the target.   1. Select the last three columns and click on the delete icon. |  |
| 1. Map the three unmapped columns as seen in the screenshot. |  |
| 1. Click on the Back icon to go back to the Data Transform editor. |  |
| 1. Click on the Back icon to go back to the graph. |  |
| 1. Add the Workflow Terminator to the graph. |  |
| 1. Connect the output port of the Data Transform operator to the Workflow Terminator operator.   This ensures the proper termination of the pipeline. |  |
| 1. Click on the Save icon in the toolbar. |  |
| 1. Set the name to teched.xx.bigdata   +enterprisedata |  |
| 1. Click on the Run icon in the toolbar to execute your pipeline. |  |
| 1. Congratulations! You’ve successfully completed the exercise.   To recap, you created a pipeline that executes the sentiment analysis pipeline, combines the results of that pipeline with product data from an HDFS file and loaded the results in SAP Vora. |  |

# Exercise3 – chart the results

In this final exercise, you will create a chart in SAP Vora to visualize the results of executing both pipelines. The main value of the data is the polarity of each review, which indicates if the review is positive (>0 and <=1 ), neutral (value 0) or negative (<0 and >= – 1)

| Explanation | Screenshot |
| --- | --- |
| 1. From the SAP Data Hub launchpad, click on the Vora Tools tile. |  |
|
|
|
| 1. Double click on the default\taXX schema where xx is your assigned group number. |  |
| 1. Right click on the Products table and select Data Preview. |  |
| 1. Click on CHARTS. |  |
| 1. Set NAME as the X axis   This will ensure you see data at the product level. You could also choose category.   1. Add REVIEW\_POLARITY to the Y axis and select AVG.   This will display the average polarity per product.   1. Add REVIEW\_POLARITY to the Y axis and select MIN.   This will display the lowest polarity (negative sentiment) per product.   1. Add REVIEW\_POLARITY to the Y axis and select MAX.   This will display the highest polarity per product. |  |
| 1. Looking at the overall chart you can see which products had the best and worst average ratings. When you hover over the chart, you can see the specific average, minimum and maximum values for the current product. In this case Notebook Basic 15.   Feel free to change the Y axis columns to see if you can gain more insights. |  |
| 1. Congratulations! You have successfully completed the exercise. |  |