Exercise #1:

HOW TO ANALYZE MACHINE AND SENSOR DATA

INTRODUCTION

This tutorial describes how to refine data from heating, ventilation, and air conditioning (HVAC) systems using the Hortonworks Data Platform, and how to analyze the refined sensor data to maintain optimal building temperatures.

SENSOR DATA

A sensor is a device that measures a physical quantity and transforms it into a digital signal. Sensors are always on, capturing data at a low cost, and powering the "Internet of Things."

POTENTIAL USES OF SENSOR DATA

Sensors can be used to collect data from many sources, such as:

- To monitor machines or infrastructure such as ventilation equipment, bridges, energy meters, or airplane engines. This data can be used for predictive analytics, to repair or replace these items before they break.
- To monitor natural phenomena such as meteorological patterns, underground pressure during oil extraction, or patient vital statistics during recovery from a medical procedure.

In this tutorial, we will focus on sensor data from building operations. Specifically, we will refine and analyze the data from Heating, Ventilation, Air Conditioning (HVAC) systems in 20 large buildings around the world.

OVERVIEW

To refine and analyze HVAC sensor data, we will:

- Download and extract the sensor data files.
- Load the sensor data into the Hortonworks Sandbox.
- Run two Hive scripts to refine the sensor data.
- Access the refined sensor data with Apache Zeppelin.
- Visualize the sensor data using Apache Zeppelin.

DOWNLOAD AND EXTRACT THE SENSOR DATA FILES

- You can download the sample sensor data contained in a compressed (.zip) folder here:
- SensorFiles.zip
- Save the SensorFiles.zip file to your computer, then extract the files. You should see a SensorFiles folder that contains the following files:
- HVAC.csv contains the targeted building temperatures, along with the actual (measured)

building temperatures. The building temperature data was obtained using Apache Flume. Flume can be usedas a log aggregator, collecting log data from many diverse sources and moving it to a centralized data store. In this case, Flume was used to capture the sensor log data, which we can now load into the Hadoop Distributed File System (HFDS). For more details on Flume, refer to Tutorial 13: Refining and Visualizing Sentiment Data

• building.csv – contains the "building" database table. Apache Sqoop can be used to

transfer this type of data from a structured database into HDFS.

LAB 1 – LOAD DATA INTO HIVE

STEP 2: LOAD THE SENSOR DATA INTO THE HORTONWORKS SANDBOX

- Navigate to the ambari login by going to the web address http://localhost:8080
- Login with the username maria_dev and password maria_dev .

Head on over to the Hive view using the dropdown menu in the top-right corner.



Then use the Upload Table tab to upload the two csv files contained

```
within SensorFiles.zip
```

| ł | Hive | Query | Saved Queries | History U | DFs Upload |
|----|------------|-------------|---------------|-----------|------------|
| Ch | noose File | No file cho | osen | | - 1 |

When uploading the two tables we'll need to change a few things.

For HVAC.csv

- Change the table name to hvac_raw
- Change the name of the Date column to date_str
- Change the type of the date_str column from DATE to STRING

| Hive Query Saved | Queries History UDFs Upload Table | | |
|---|-----------------------------------|------------------------|--------------|
| Choose File HVAC.csv Database : default date_str 2 | Table Name : hvac_raw Time | Is First Row Header? : | Upload Table |
| STRING 3 | ▼ STRING | • INT | ▼ INT |
| 6/1/13 | 0:00:01 | 66 | 58 |
| 6/2/13 | 1:00:01 | 69 | 68 |
| 6/3/13 | 2:00:01 | 70 | 73 |
| 6/4/13 | 3:00:01 | 67 | 63 |

For buildings.csv

Change the table name to building_raw

| Hive Query Saved Queries Hi | story UDFs Upload Table | | |
|---|------------------------------|------------------------|--------------|
| Choose File building.csv Database : default | Table Name : building_raw | Is First Row Header? : | Upload Table |
| BuildingID | BuildingMgr | BuildingAge | HVACproduct |
| INT | STRING | • INT | ▼ STRING |
| 1 | M1 | 25 | AC1000 |
| • | 110 | ~~ | FUNCTO |

• Now that we have both tables loaded in, we want to get better performance in Hive, so we're going to create new tables that utilize the highly efficient ORC file format. This will allow for faster queries when our datasets are much larger.

• Execute the following query to create a new table hvac that is stored as an ORC file.

| CREATE TABLE hvac STORED AS ORC AS SELECT * FROM hvac_raw; | | | | | | | |
|--|---------------|--|--|--|--|--|--|
| | | | | | | | |
| Query Editor | ×* | | | | | | |
| Worksheet | | | | | | | |
| 1 CREATE TABLE hvac STORED AS ORC AS SELECT * FROM HVAC_stage; | | | | | | | |
| | | | | | | | |
| Execute Explain Save as Kill Session | New Worksheet | | | | | | |

• Repeat the previous step, except this time we will make a table for buildings .

CREATE TABLE buildings STORED AS ORC AS SELECT * FROM building_raw;

| Query Editor | 2 |
|--|---------------|
| Worksheet * | |
| 1 CREATE TABLE buildings STORED AS OF AS SELECT * FROM buildings_stage; 2 | |
| | |
| | |
| Execute Explain Save as Kill Session | New Worksheet |

STEP 3: RUN TWO HIVE SCRIPTS TO REFINE THE SENSOR DATA

We will now use two Hive scripts to refine the sensor data. We hope to accomplish three goals with this data:

- Reduce heating and cooling expenses.
- Keep indoor temperatures in a comfortable range between 65-70 degrees.
- Identify which HVAC products are reliable, and replace unreliable equipment with those models.

- First, we will identify whether the actual temperature was more than five degrees different from the target temperature.
- Create a new worksheet in the Hive view and paste the following Hive query into your window.

```
CREATE TABLE hvac_temperatures as
select *, targettemp - actualtemp as temp_diff,
IF((targettemp - actualtemp) > 5, 'COLD',
IF((targettemp - actualtemp) < -5, 'HOT', 'NORMAL'))
AS temprange,
IF((targettemp - actualtemp) > 5, '1',
IF((targettemp - actualtemp) < -5, '1', 0))
AS extremetemp from hvac;
```

- This query creates a new table hvac_temperatures and copies data from the hvac table
- After you paste the query use **Execute** to create the new table.

| Database Explorer | C | Query Editor | ×* | |
|-------------------|---|---|---------------|----------|
| default | • | Worksheet | | 0 |
| | | <pre>1 CREATE TABLE hvac_temperatures as select *, 2 targettemp - actualtemp as temp diff.</pre> | | SQL |
| Search tables | | <pre>3 IF((targettemp = actualtemp) > 5, 'COLD', 4 IF((targettemp = actualtemp) < -5, 'HOT', 'NORMAL'))</pre> | | \$ |
| Databases | | 5 AS temprange, 6 IF((targettemp - actualtemp) > 5, '1', 7 TF((targettemp - actualtemp) > 5, '1', | | 1. |
| S default | | 8 AS extremetemp from hvac; | | |
| | | | | 8 |
| ⊞ hvac | | | | TE7 |
| hvac_stage | | | | 8 |
| sample 07 | | | | \simeq |
| ⊞ sample_08 | | | | |
| ≣ xademo | | Execute Explain Save as Kill Session | New Worksheet | |
| | | | | |
| | | 100% | | |
| | | | | |
| | | Query Process Results (Status: SUCCEEDED) | Save results | |

• On the Query Results page, use the slider to scroll to the right. You will notice that two new attributes appear in the hvac_temperatures table.

The data in the **temprange** column indicates whether the actual temperature was:

- NORMAL within 5 degrees of the target temperature.
- **COLD –** more than five degrees colder than the target temperature.
- **HOT –** more than 5 degrees warmer than the target temperature.

If the temperature is outside of the normal range, extremetemp is assigned a value of 1; otherwise its value is 0.

| Query Process Results (Status: Succeeded) Save results | | | | | | | |
|--|-----------------------------|-----------------------------|-------------------------------|--|--|--|--|
| Logs Result | S | | | | | | |
| Filter columns | | | previous next | | | | |
| ratures.buildingid | hvac_temperatures.temp_diff | hvac_temperatures.temprange | hvac_temperatures.extremetemp | | | | |
| | 4 | NORMAL | 0 | | | | |
| | 10 | COLD | 1 | | | | |
| | 2 | NORMAL | 0 | | | | |
| | 2 | NORMAL | 0 | | | | |
| | 7 | COLD | 1 | | | | |
| | 2 | NORMAL | 0 | | | | |
| | -11 | НОТ | 1 | | | | |
| | 7 | COLD | 1 | | | | |
| | 1 | NORMAL | 0 | | | | |
| | 10 | COLD | 1 | | | | |
| | 1 | NORMAL | 0 | | | | |
| | -10 | НОТ | 1 | | | | |
| 4 | -7 | НОТ | 1 | | | | |
| | 8 | COLD | 1 | | | | |
| | 10 | COLD | 1 | | | | |
| | -6 | НОТ | 1 | | | | |
| | -10 | нот | 1 | | | | |

• Next we will combine the **hvac** and **hvac_temperatures** data sets. Create a new worksheet in the hive view and use the following query to create a new

```
tablehvac_buildingthat contains data from thehvac_temperaturestable andthebuildingstable.
```

```
create table if not exists hvac_building
as select h.*, b.country, b.hvacproduct, b.buildingage, b.buildingmgr
from buildings b join hvac_temperatures h on b.buildingid = h.buildingid;
```

• Use **Execute** to run the query that will produce the table with the intended data.

| Query Editor | ~ | |
|---|----|-----|
| Worksheet × Worksheet (4) × | | 0 |
| <pre>1 create table if not exists hvac_building 2 as select h.*, b.country, b.hvacproduct, b.buildingage, b.buildingmgr</pre> | | SQL |
| <pre>3 from buildings b join hvac_temperatures h on b.buildingid = h.buildingid;</pre> | | ۰ |
| | | |
| | | °0 |
| R. | | TEZ |
| | | |
| Execute Explain Save as Kill Session New Workshe | et | |

• After you've successfully executed the query, use the database explorer to load a sample of the data from the new hvac_building table.

| Database Explorer | C | Query Editor | | | | 2 | |
|---------------------------|---|----------------------------|--------------------|--------------------------|--------------------------|---------------|-----|
| default | • | Worksheet × hvac_b | uilding sample X | | | | 0 |
| Quarte tablas | | 1 SELECT * FROM hva | c_building LIMIT | 100; | | | SQL |
| Search tables | | | | | | | \$ |
| Databases | | | | | | | |
| efault | | | | | | | 90 |
| ⊞ buildings_stage | | | | | | | TEZ |
| ⊞ hvac ⊞ hvac_building | | | | | | | |
| hvac_stage hvac_stage | | | | = | | | - |
| sample_07 sample_08 | | Execute Explain Save | as Kill Session | | | New Worksheet | |
| ≘ xademo | | | | | | | |
| | | Query Process Results (Sta | atus: Succeeded) | | | Save results | |
| | _ | Logs Results | | | | | |
| | | Filter columns | | | | previous next | |
| | | hvac_building.recorddate | hvac_building.time | hvac_building.targettemp | hvac_building.actualtemp | hvac_building | |
| | | 6/1/13 | 0:00:01 | 66 | 58 | 13 | |
| | | 6/2/13 | 1:00:01 | 69 | 68 | 3 | |
| | | 6/3/13 | 2.00.01 | 70 | 73 | 17 | |

Now that we've constructed the data into a useful format, we can use different reporting tools to analyze the results.

ACCESS THE REFINED SENSOR DATA WITH APACHE ZEPPELIN

Apache Zeppelin makes data reporting easy on Hadoop. It has direct connections to Apache Spark and Hive in your cluster and allows you to create visualizations and analyze your data on the fly.

To start you're going to need to open up the <u>Apache Zeppelin view</u> in Ambari. Start by navigating back to the <u>Ambari Dashboard</u> at <u>http://localhost:8080</u>

• Use the dropdown menu to open the Zeppelin View.

| 🚕 Ambari Sand | lbox 10 ops 0 alerts | | Dashboard Service | s Hosts Alerts Adı | min | admin - |
|---|---|-------------------------------|---|--------------------|-----|--|
| HDFS MapReduce2 YARN Tez | Metrics Heatmaps Metric Actions - HDFS Disk Usage | Config History DataNodes Live | HDFS Links | Memory Usage | N | YARN Queue Manager HDFS Files Local Files Hive Pig |
| Hive HBase Pig Sqoop | 4256 | 1/1 | NameNode Secondary NameNode 1 DataNodes More • | No Data Available | | Storm Sez View Zeppelin |
| CozieCooKeeper | CPU Usage | Cluster Load | NameNode Heap | NameNode RPC | N | ameNode CPU WIO |

- From here we're going to need to create a new Zeppelin Notebook.
- Notebooks in Zeppelin is how we differentiate reports from one another.
- Hove over Notebook. Use the dropdown menu and Create a new note.

| 🛛 🚕 Ambari 🛛 Sandbox 🚽 | 0 ops 0 alerts | Dashboard | Services | Hosts | Alerts | Admin | 🛓 admin 👻 |
|--|--|--------------------------------|----------|-------|--------|-------|---------------|
| 🥖 Zeppelin 🄇 | Notebook - Interpreter | | | | | | Connected |
| Welcome to | + Create new note | | | | | | |
| Zeppelin is web-based noteb You can make beautiful data | Australian Dataset (Hive example) Australian Dataset (SparkSQL example) | th SQL, code and even more! | | | | | |
| Notebook | Driver Risk Factor IoT Data Analysis (Keynote Demo) | h Zeppelin documentation | | | | | |
| AON Demo Australian Dataset (H | Phoenix demo Pyspark test | to help us to improve Zeppelin | | | | | |
| Australian Dataset (Driver Risk Factor | Zeppelin Tutorial | on are welcome! | | | | | |

• Name the note HVAC Analysis Report and then Create Note.

| Noteboo | ok - Interp | reter | | |
|------------------------------|-------------|---|-------------|--|
| | Create new | note | × | |
| Zep | Note Name | | | |
| ook that ena | HVAC Analys | is Report | | |
| driven, inter | | | | |
| | | Community | Create Note | |
| live example) parkSQL exa |) ample) | Please feel free to help us to improve Zeppelin, Any contribution are welcome! | | |

- Head back to the Zeppelin homepage.
- Use the **Notebook** dropdown menu to open the new notebook **HVAC Analysis Report**.

| eter |
|--------------------------------------|
| |
| |
| Imple) QL example) th SQL, code a |
| |
| h Zeppelin docı |
| Demo) |
| te hele ve te iv |
| are welcome |
| |
| Mailing list |
| |
| Github |
| |

- Zeppelin integrates with Hadoop by using things called *interpreters*.
- In this tutorial we'll be working with the Hive interpreter to run Hive queries in Zeppelin, then visualize the results from our Hive queries directly in Zeppelin.

• To specify the Hive interpreter for this note, we need to put <a>hive at the top of the note. Everything afterwards will be interpreted as a Hive query.



 Type the following query into the note, then run it by clicking the Run arrow or by using the shortcut Shift+Enter.

```
%hive
select country, extremetemp, temprange from hvac_building
"`
```



• After running the previous query we can view a chart of the data by clicking the chart button located just under the query.

| 🔵 Zeppelin | Notebook - Interpreter | Con |
|--------------------------------------|---------------------------------|----------------|
| VAC Analysis F | eport DXWD2 0 | ⑦ \$ det |
| %hive | | FINISHED D X 🗐 |
| select country, extremeter | p, temprange from hvac_building | |
| country | extremetemp | temprange |
| Finland | 1 | COLD |
| Egypt | 0 | NORMAL |
| Indonesia | 0 | NORMAL |
| Israel | 0 | NORMAL |
| Brazil | 1 | НОТ |
| Finland | 1 | COLD |
| France | 1 | COLD |
| Turkey | 0 | NORMAL |
| Mexico Took 5 seconds. (outdated) | 0 | NORMAL |
| | | |

• Click **settings** to open up more advanced settings for creating the chart. Here you can experiment with different values and columns to create different types of charts.



- Arrange the fields according to the following image.
- Drag the field temprange into the groups box.
- Click **SUM** on extremetemp and change it to **COUNT**.
- Make sure that country is the only field under **Keys**.

| %hive | | | FINISHED ▷ 💥 🗐 🐵 |
|---|--------------|-------------------|------------------|
| select country, extremetemp, temprange from I | wac_building | | |
| 🖽 🔟 🔄 🕍 🖄 settings 🔺 | | | |
| All fields: | | | |
| country extremetemp temprange | | | |
| Keys | Groups | V-ues | |
| country × | temprange × | extremetemp COUNT | \mathbf{P} |
| | | | |
| | | | |
| | | | |
| | | | |

Awesome! You've just created your first chart using Apache Zeppelin.

- From this chart we can see which countries have the most extreme temperature and how many **NORMAL** events there are compared to **HOT** and **COLD**.
- It could be possible to figure out which buildings might need HVAC upgrades, and which do not.



- Let's try creating one more note to visualize which types of HVAC systems result in the least amount of extremetemp readings.
- Paste the following query into the blank Zeppelin note following the chart we made previously.

```
%hive
select hvacproduct, extremetemp from hvac_building
"`
```

• Now use **Shift+Enter** to run the note.

| %hive select hvacproduct, extremetemp from hvac_building | k | FINISHED Þ 💥 🗌 🐵 |
|--|---|------------------|
| 1 | | READY D 💥 🗐 🐵 |

- Arrange the fields according to the following image so we can recreate the chart below.
- Make sure that hvacproduct is in the Keys box.
- Make sure that extremetemp is in the **Values** box and that it is set to **COUNT**.



• Now we can see which HVAC units result in the most extremetemp readings. Thus we can make a more informed decision when purchasing new HVAC systems.

Apache Zeppelin gives you the power to connect right to your Hadoop cluster to quickly obtain results from the data inside of Hadoop without having to export data to any other sources.

It's also important to note that Zeppelin contains many, many interpreters that can be utilized to obtain data in a variety of ways.

One of the default interpreters included with Zeppelin is for Apache Spark. With the popularity of Apache Spark rising, you can simply write Spark scripts to execute directly on Apache Zeppelin to obtain results from your data in a matter of seconds.