

PI World 2019 Lab

Introduction to Data Science for PI System Data
for PI Professionals



OSIssoft, LLC
1600 Alvarado Street
San Leandro, CA 94577 USA
Tel: (01) 510-297-5800
Web: <http://www.osissoft.com>

© 2019 by OSIssoft, LLC. All rights reserved.

OSIssoft, the OSIssoft logo and logotype, Analytics, PI ProcessBook, PI DataLink, ProcessPoint, Asset Framework (AF), IT Monitor, MCN Health Monitor, PI System, PI ActiveView, PI ACE, PI AlarmView, PI BatchView, PI Vision, PI Data Services, Event Frames, PI Manual Logger, PI ProfileView, PI WebParts, ProTRAQ, RLINK, RtAnalytics, RtBaseline, RtPortal, RtPM, RtReports and RtWebParts are all trademarks of OSIssoft, LLC. All other trademarks or trade names used herein are the property of their respective owners.

U.S. GOVERNMENT RIGHTS

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the OSIssoft, LLC license agreement and as provided in DFARS 227.7202, DFARS 252.227-7013, FAR 12.212, FAR 52.227, as applicable. OSIssoft, LLC.

Published: March 26, 2019

Table of Contents

Contents

Table of Contents.....	3
Learning Objectives.....	4
PI System Software Components.....	5
Part 1 – Introduction.....	6
1.1 Introduction to Data Science concepts – CRISP DM Methodology	6
1.2 Business Objective	9
Part 2 – Data understanding	11
2.1 What data is available?	11
2.1 Data Understanding through PI Vision	13
2.2 Explore generated Event Frames	18
Part 3 – Exploratory Data Analysis.....	20
3.1 Publish dataset using the PI Integrator for Business Analytics.....	20
3.2 Data Exploration using Microsoft Power BI Desktop.....	25
3.2.1 Loading the Data in Power BI.....	25
3.2.2 Creating Reports for Data Exploration.....	27
Part 4 – Modeling and Evaluation.....	43
4.1 Building a model in Orange.....	43
4.1.1 Loading the dataset	44
4.1.2 Filtering, Feature Engineering and Feature selection	47
4.1.3 Splitting the data into Training and Test sets	52
4.1.3 How to choose the right model	53
4.1.4 Results Evaluation	57
Part 5 – Deployment	63
Appendix - Publish Data with PI Web API	66
Import Power Consumption Data in Jupyter Notebooks.....	66
Save the Date!.....	69

Learning Objectives

In this lab you will be introduced to the key concepts of Data Science. If you are a PI professional and you are already familiar with the basic PI Tools, this lab will get you one step further in the process of analyzing your data and getting value out of it. The lab will go through all the steps of a Data Science project cycle, from defining the Business Objective, to the part of doing some advanced Business Intelligence analysis on your data. We will go through the process of getting data from PI into other advanced analytics platforms like R, Python, Microsoft Power BI and Azure ML. The topics will also include the process of cleaning your data and bringing it into shape, as well as focus on the exploratory data analysis part, where you will discover interesting relationships in your data. At the end of this lab, you will have gained knowledge on the major Data Science techniques that will get you started in getting more value out of your PI Data.

More specifically, the lab is structured in the following sections:

- Part 1 – Introduction
 - Introduction to Data Science concepts
 - Introduce the Business Objective and available data
- Part 2 – Data Understanding
 - Explore available data in PI
 - Understand the project scope through PI Vision displays
 - Explore generated Event Frames
- Part 3 – Exploratory Data Analysis
 - Publish dataset using the PI integrator for Business Analytics
 - Bring the data in Power BI
 - Data Exploration using Power BI
- Part 4 – Modelling and Evaluation
 - Bringing the data in Orange
 - Feature Engineering
 - Splitting the data into training and test sets
 - How do we pick the right model
 - Evaluation
- Part 5 – Deployment
 - Deploy the model and store the predictions in PI

PI System Software Components

The VM (virtual machine) used for this lab has the following PI System software components installed:

Software	Version
PI Data Archive	2018
PI Asset Framework (PI AF) server	2018
PI Asset Framework (PI AF) client (PI System Explorer)	2018
PI Analysis Service	2018
PI Vision	2017 R2
PI Web API	2018
PI Integrator for Business Analytics	2018 R2 Advanced

Part 1 – Introduction

1.1 Introduction to Data Science concepts – CRISP DM Methodology

What this lab is: This lab is an introduction to Data Science concepts, for people who are familiar with using the basic BI tools. The scope of the lab is to introduce you to basic Data Science concepts and techniques, by going through the steps of a Data Science project example, from the formation of the Business Objective to Model Building and evaluation. The aim is to empower you in the process of engaging in Data Science initiatives.

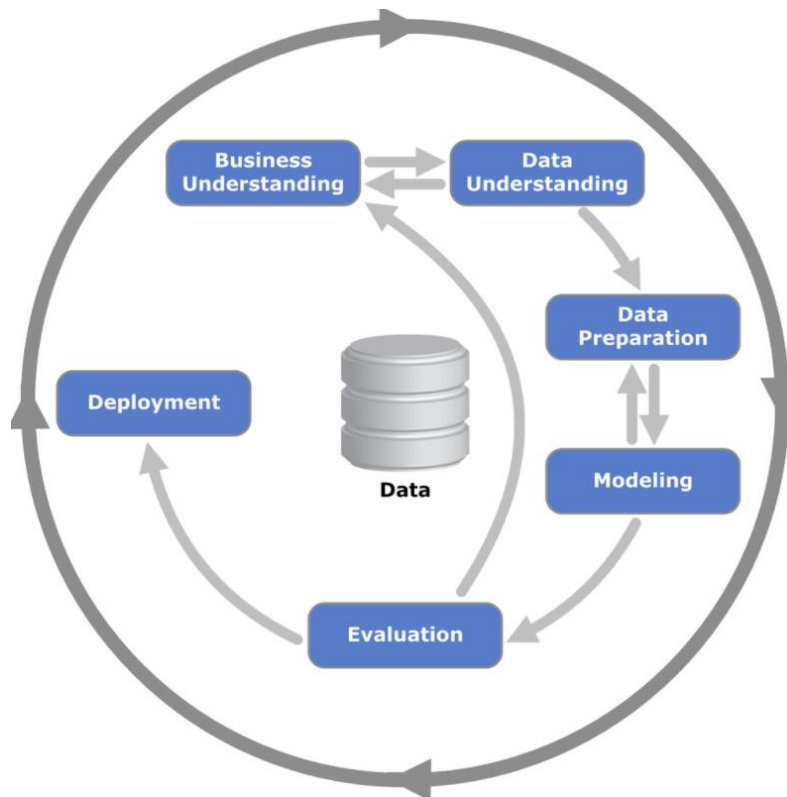
What this lab is not: This lab does not aim to serve as an alternative to the many online Data Science courses that exist and cannot be used as a full and only training resource for someone who wants to get into the field of Data Science.

This lab should probably start by giving a definition of what is Data Science. However, after giving it a lot of thought, I decided not to list one of the plenty definitions that have been used to describe this concept. A simple google search will give you numerous resources that aim to describe the term Data Science; some using scientific definitions, some using more simple terms, some trying to break down the skills required for a person to be considered as a Data Scientist, etc.

Instead, I decided that I will just start by saying that by the end of this lab, you should be able to understand some of the main Data Science concepts and techniques and you should be motivated to go back to your jobs and start using those familiar BI Datasets that you have available, in order to discover “hidden” value that might not be directly visible by just looking into everyday reports.

This is going to be done by going through an Example Data Science project, following a specific set of steps that belong to a methodology called **CRISP DM**.

CRISP DM stands for Cross-industry standard process for data mining and it provides an overview of the lifecycle of a Data Science project. It involves six main phases and it's important to remember that relationships could exist between any of the tasks at any phase of the project, depending on the goals, the background, and the specific use case. This is better illustrated on the following picture:



https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining

The outer circle of the above figure shows the cyclical nature of data science itself. A Data science project does not end once a solution is deployed. The lessons learned during the process and from the deployment of the solution can trigger new, often more-focused business questions.

Let's now briefly describe the steps of this process:

Business understanding

This initial phase focuses on understanding the project objectives and requirements from a business perspective, converting this knowledge into a clear problem definition and a preliminary plan to achieve the objectives. This part includes a lot of interaction with the stakeholders of the project and the Subject Matter Experts (SMEs), whose input is invaluable and of critical importance to the project.

Data understanding

The data understanding (or Data Exploration, or Exploratory Data Analysis) phase starts with initial data collection and exploration and aims to enable you to become familiar with the available data. In this step potential data quality problems can be identified and there is also a first assessment of how much data there is and if it seems to be adequate enough in order to achieve our objective. Furthermore this is the phase where you discover the first insights into the data, and/or detect interesting subsets to form hypotheses regarding hidden information that can help you later in the model selection and building process.

Data preparation

The data preparation step includes all the activities that are required in order to construct the final dataset and bring it into a shape that is ready to be consumed by a machine learning model. The data preparation activities normally take place multiple times within the course of the project and they are not performed in any specific order. Data preparation includes tasks such as handling missing data, cleaning outliers, feature selection, as well as potential data transformation and feature construction.

Modeling

During this phase, various modeling techniques are selected and applied to the dataset, and their parameters are tuned in order to achieve the best possible outcome. In most cases there are several different models that can be applied to the same data science problem type, although some techniques have specific requirements on the form of data and therefore, going back to the data preparation phase is often necessary, as mentioned previously.

Evaluation

At this stage in the project, you have built a model (or models) that appears to have high quality from a data analysis perspective. Before proceeding to the final deployment of the model, it is important to thoroughly evaluate it and review the steps executed to create it, to be certain the model properly achieves the business objectives. At the end of this phase, a decision on the use of the results should be reached and it is quite common from this stage to go back to step 1, the Business Understanding, to discuss with the stakeholders and SMEs, in order to make sure that the business objective is addressed and check if there is any need to refine the objective. So don't be discouraged if from this stage instead of going to deployment, you go back to step 1 and refine your objectives. This is actually very common.

Deployment

Creation of the model is generally not the end of the project. Even if the purpose of the model is to increase knowledge of the data, the knowledge gained will need to be organized and presented in a way that it can be used. It often involves applying "live" models within an organization's decision making processes. Depending on the requirements, the deployment phase can be as simple as generating a report or as complex as implementing a repeatable data mining process across the enterprise.

So now that we know the steps that we need to follow for our data science project, let's get into it by first understanding our Business Objective

1.2 Business Objective

In this lab, we are going to look into a use case of using data to improve the energy efficiency of a building, by reducing the energy that is consumed for cooling. More specifically, we will look into optimizing the daily startup of the individual **Variable Air Volume Cooling** units (**VAVCO**).

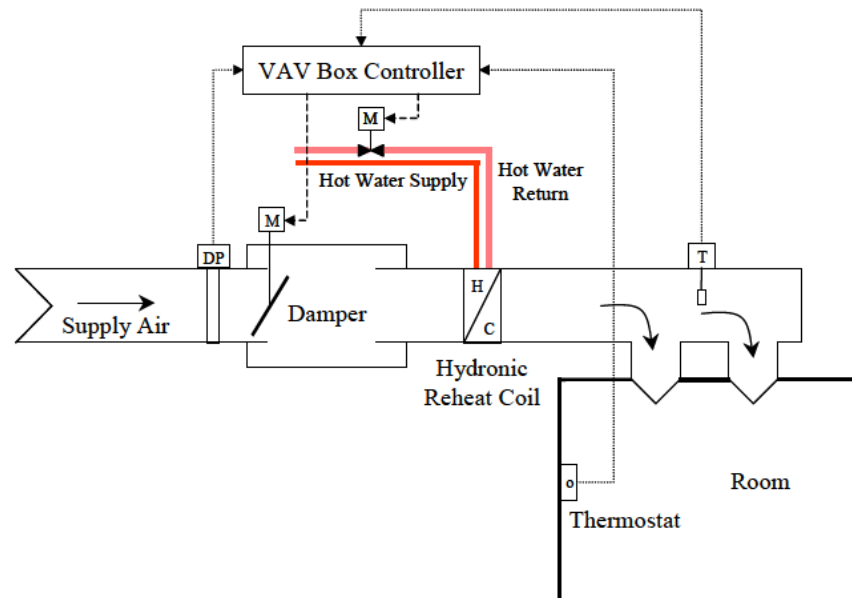


Figure 1: Schematic diagram of a single duct pressure-independent VAV box with hydronic reheat

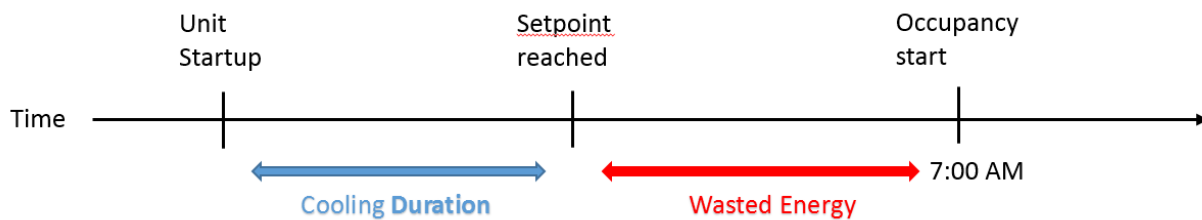
During the course of a day, the VAVCO unit's operating conditions change continuously as the room temperature rises and falls, along with changing relative humidity, changing thermostat set-points, building occupancy level, and others. The BMS control system adjusts the supply air flow rate, damper position etc. to provide the necessary cooling to the rooms and ensure tenant comfort.

Furthermore, there is an occupancy schedule that has been set up, to define the hours when people are in the building and the temperature needs to be at comfort levels. During those hours, the individual VAVCO units are working to bring the temperature to the desired setpoint. This occupancy schedule has been set up to the hours of **7 AM to 7 PM** in this case.

This is the daily pattern of the VAVCO units operation:

- Turn-on at some point in the morning, before 7 AM, and try to bring the room temperature at setpoint by 7 AM
- Operate to keep the room temperature at setpoint during the course of the occupancy schedule
- Shut-down at 7 PM when the occupancy schedule ends and the temperature doesn't need to be at setpoint anymore

After discussing with our SME's, we have found out that one of their objectives is to optimize the startup of the units. What that means is that we basically need to make sure that room temperature is at setpoint, as close as possible to 7 AM. If we reach the setpoint too early then energy is being wasted to keep the temperature at setpoint when the building is still unoccupied, while of course if the setpoint is not reached by 7 AM, this will be a discomfort for the people that have already started coming into the building.



In conclusion, our objective is to be able to predict how much time will take to reach the setpoint depending on the current conditions, so that we can move the startup of the VAVCO units as close as possible to 7 AM.

Sensor data available from the VAVCO units, as part of the BMS (building management system) are:

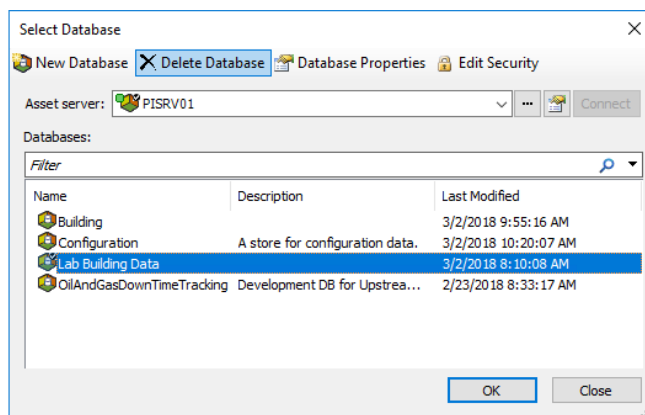
- Outside air temperature
- Relative Humidity
- Room temperature
- Damper position
- Supply air flow
- ...

Part 2 – Data understanding

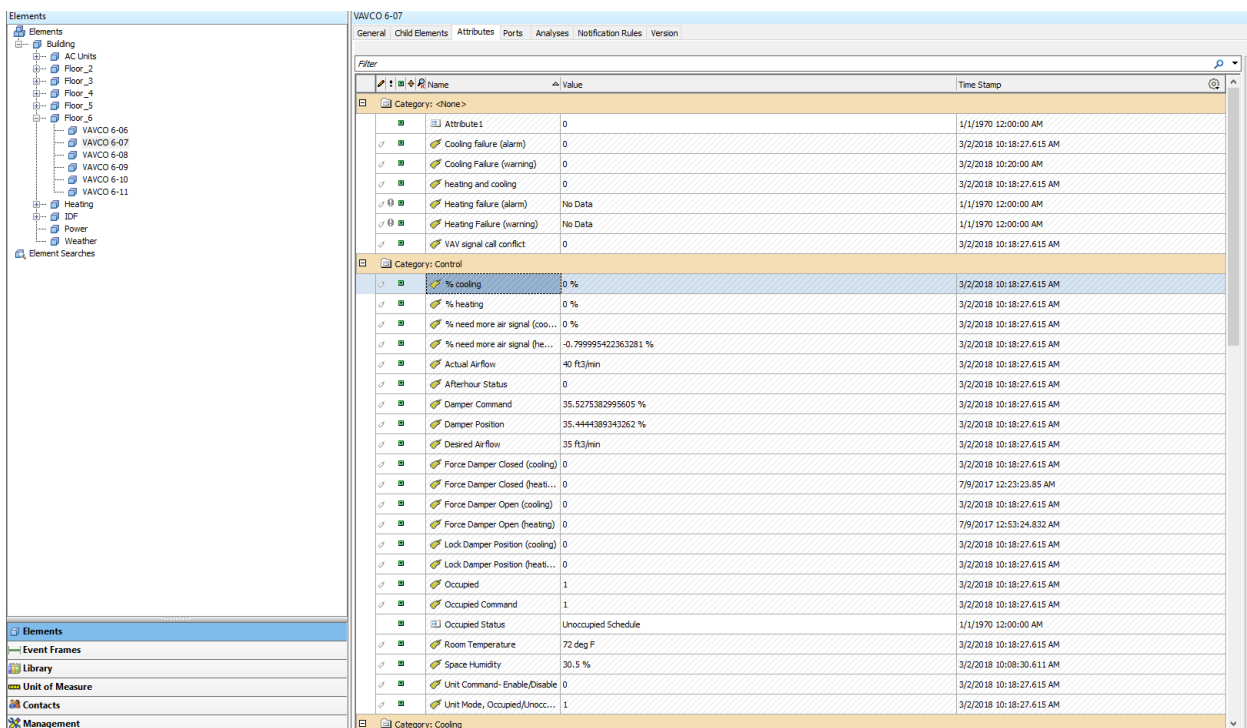
2.1 What data is available?

Now that we have identified our Business Objective, it's time to start what data is available to us in order to achieve this objective.

1. Open PI System Explorer (PSE) and make sure that you are connected to the **Lab Building Data** database.



2. Navigate through the AF hierarchy to **Building > Floor 6 > VAVCO 6-06** and look at the attributes of this element.



As you can see, our VAVCO units are organized by Floor and each of the VAVCO units has a list of attributes associated with it, which describe its current behavior. You can notice attributes like **% Cooling, Actual Airflow, Room Temperature**, etc and we will be focusing on a subset of those attributes in order to predict the duration that each unit needs in order to reach the setpoint temperature.

Notice that there is also a **Weather** Element in our hierarchy which contains attributes associated with external weather conditions, as well as a **Power Element**, which contains information about the Power Usage of the building.

The image displays two side-by-side screenshots of a software interface, likely a data management or analysis tool. Both screenshots show a hierarchical tree of elements on the left and a detailed view of a selected element on the right.

Left Screenshot: Weather Element

- Elements Tree:** Shows a hierarchy starting with 'Building', followed by 'AC Units', 'Floor_2', 'Floor_3', 'Floor_4', 'Floor_5', and 'Floor_6'. Under 'Floor_6', there are several VAVCO units (VAVCO 6-06 to VAVCO 6-11), 'Heating', 'IDF', 'Power', and 'Weather'. The 'Weather' element is highlighted with a red box.
- Weather Detail View:** Shows a table of attributes for the 'Weather' element. The table has columns for 'Name' and 'Value'.

Name	Value
Device ID	101056
Dewpoint Temperature	38.4000091552734 deg F
Outside Air Temperature	51.5000114440918 deg F
Precipitation (ABS)	0.12399999797344208
Relative Air Pressure	1010.4000244140625
Relative Humidity Percentage	60.8000106811523 %
Wet Bulb Temperature	45.0999984741211 deg F
Wind Chill Temperature	48.5 deg F
Wind Direction	181.699996948242 °
Wind Heating Temperature	53.5000114440918 deg F
Wind Speed	11.699999809265137

Right Screenshot: Power Element

- Elements Tree:** Shows the same hierarchy as the left screenshot. The 'Power' element is highlighted with a red box.
- Power Detail View:** Shows a table of attributes for the 'Power' element. The table has columns for 'Name' and 'Value'.

Name	Value
Building kW	17.6

Now that we have an idea of what data is available, we are ready to start Exploring the data and try to find indicator attributes that will help us identify and analyze the behavior of the Cooling startup events.

2.1 Data Understanding through PI Vision

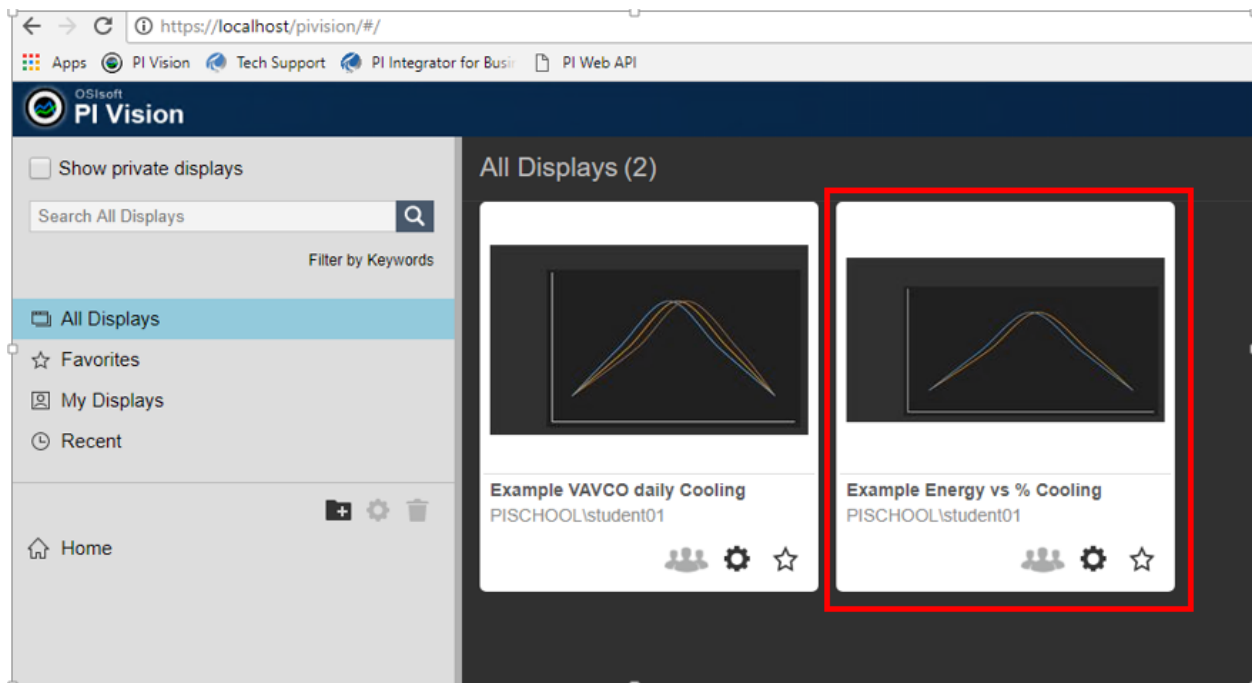
At this part of the lab, we are going to use some of our familiar PI tools in order to explore the available data and further understand the scope of our project, which is exactly what we would do initially if we were presented with this task.

Energy vs % Cooling Display

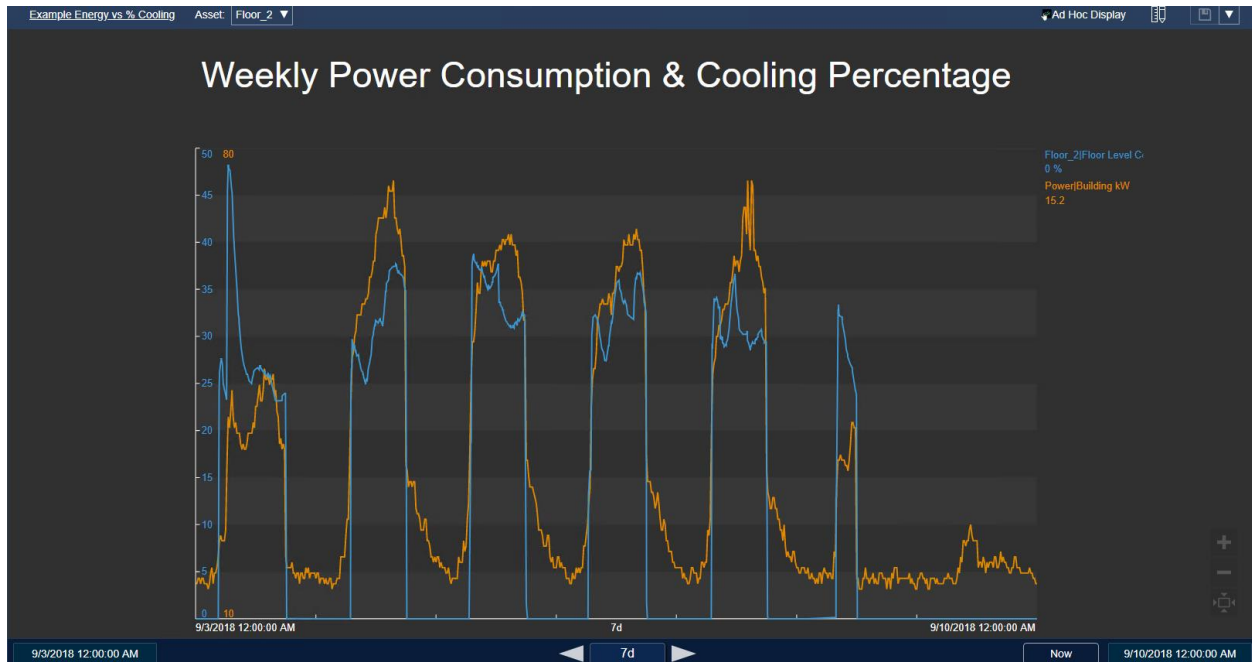
1. Open up Google Chrome from your task bar and select the PI Vision shortcut from the bookmarks menu. When you get a logon prompt, type the credentials used to log into your VM. It might take a while to load for the first time.




2. Now select the “**Example Energy vs % Cooling**” display from the **All Displays** menu.



This is a pre-built PI Vision display that looks at the Power Usage in respect to Average % Cooling, within a period of 1 week.



There are a few things that we can notice in this display:

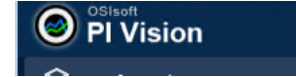
 <p>Discussion</p>	<ul style="list-style-type: none">- First of all, we can clearly see that the daily power usage is aligned with the Average % Cooling attribute which is a first indication that if we optimize our cooling process, that should have an impact on the overall energy consumption.- Furthermore, we can see that on the 6th day the Power Usage is lower than the rest of the days and on the 7th day the Power Usage drops to a constant minimum value. This is already one question which we would take back to our SMEs to get a better understanding of what we see in our data. The answer we got in that case is that the 6th day is a Saturday and the occupancy schedule is set to 7AM – 1PM on Saturdays and that on Sundays the building is shut down and there is no cooling taking place, so that's why we don't see a rise in power usage.
---	---

As a result, we already have some very useful information which we can keep in mind for our Data Analysis!

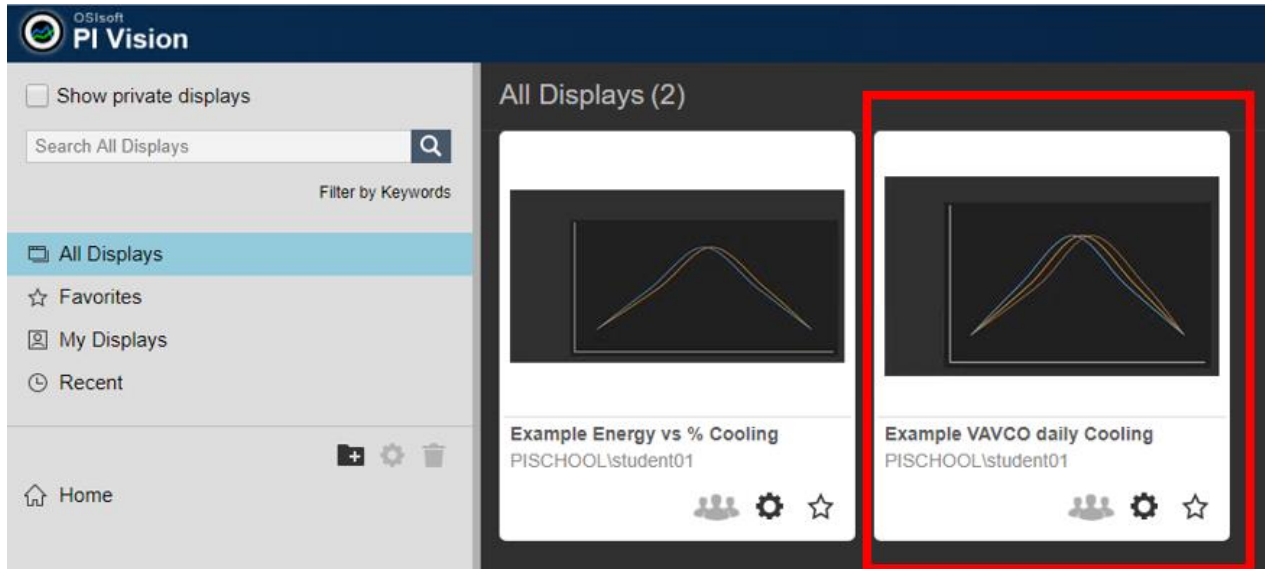
VAVCO daily Cooling Display

Now let's get back to our main goal, which is to optimize the startup of the cooling process. We are going to look into another PI Vision display showing the daily operation of a VAVCO unit and see how we can capture its startup so that we can further analyze it.

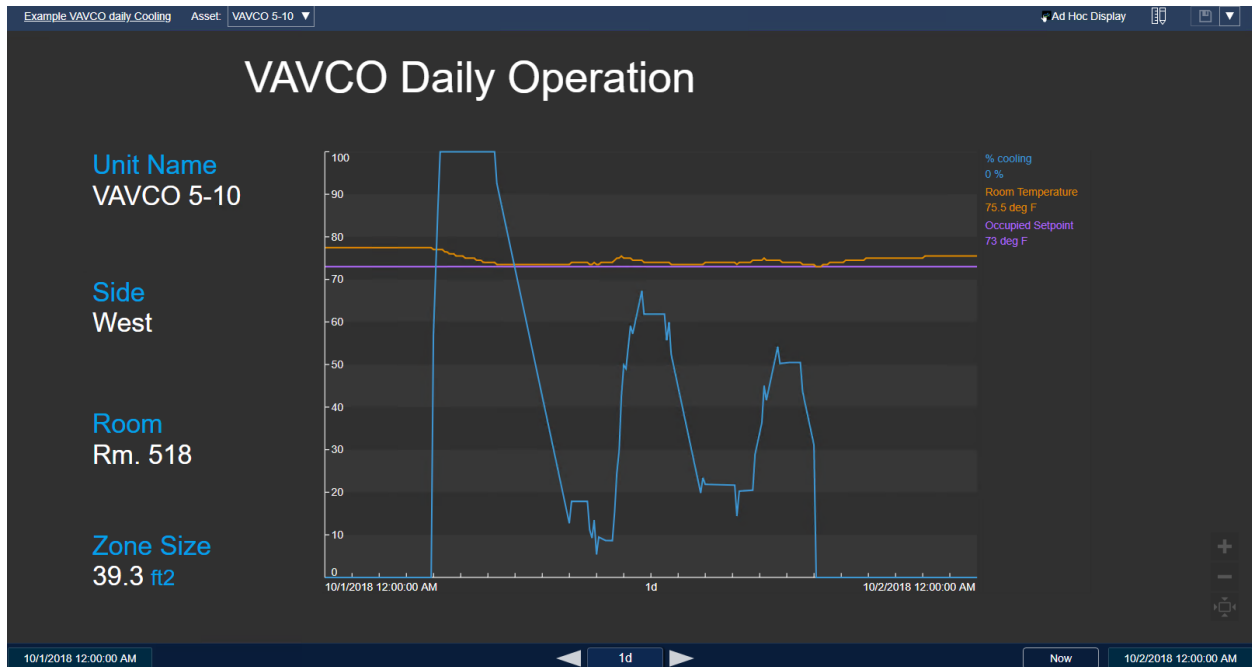
3. Select the PI Vision symbol on the top left of your browser in order to go back to the home page.



4. Let's have a look at the **Example VAVCO daily Cooling** display.




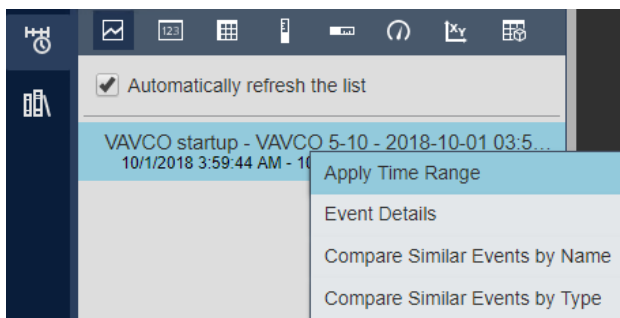
This is a pre-built PI Vision display showing a trend of the % Cooling attribute during a period of 1 day, as well as the behavior of the room temperature compared to the setpoint temperature

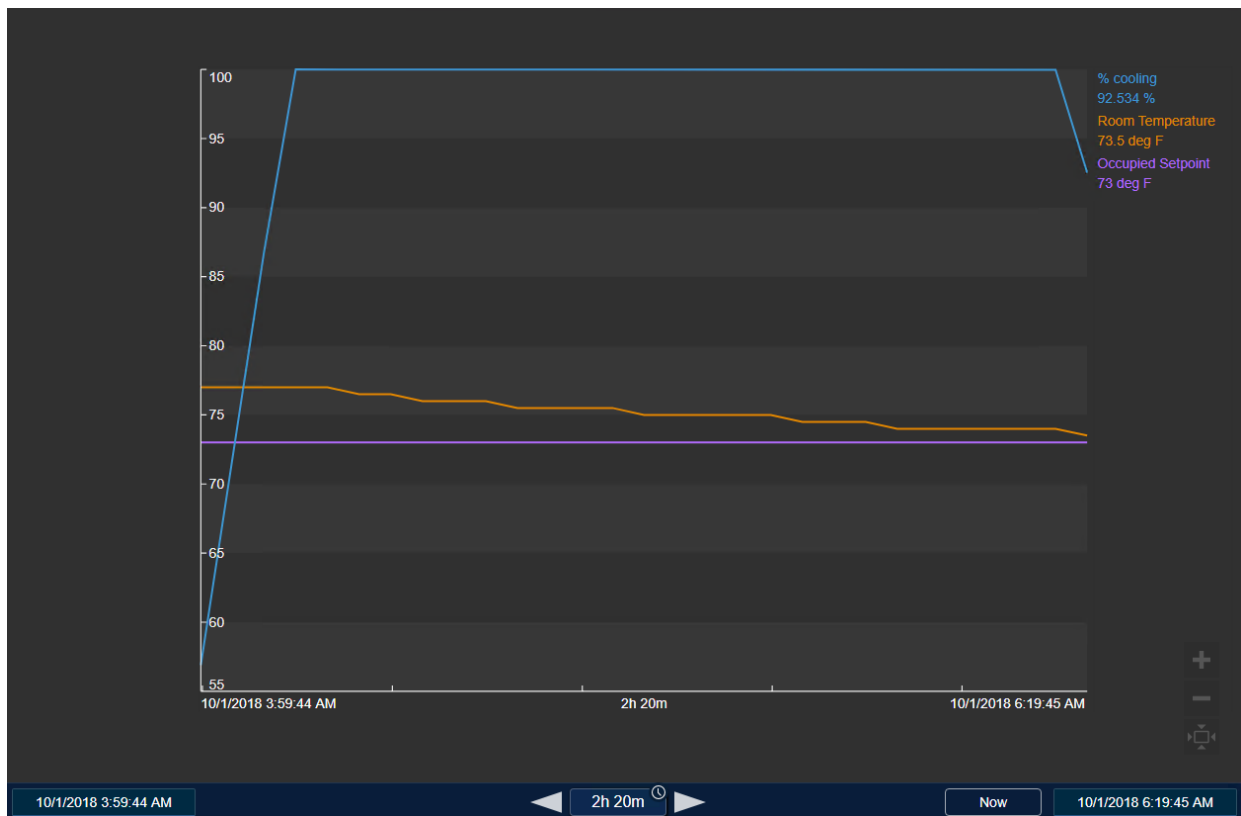


For our use case, we want to capture and analyze all the morning startup events, which can be identified by the first rise of % Cooling attribute on the left part of the above graph. Our goal is to analyze how long does this first part of the above trend takes, from the time that the % Cooling attribute rises for the first time, until the room temperature reaches the occupied setpoint temperature. If we can predict how long this startup process takes, then we should be able to optimize the startup process and make it reach the setpoint as close as possible to 7 AM, so that we are not wasting energy by cooling when there is nobody inside the building.

For that reason, we have created some Event Frames in order to capture all those startup events for all our VAVCO units. If you look on the left of the PI Vision screen under the Assets pane, you will notice a symbol that indicates that there are some event frames that have been captured.

5. Select the Event Frame symbol,  right click on the generated Event and select *Apply Time Range*. This will adjust the time range of the display to focus only on the startup event.





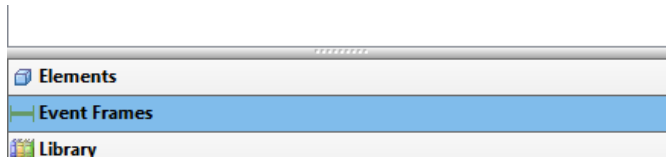
We can see in this case that the duration of the Event was 2h 20m, we can see the rise of the % Cooling attribute and the Room temperature starting from 77 degrees at the start of the event and going down to 73 at the end of our event frame. This means that in this case it took 2 hours and 20 minutes in order to get the room temperature at setpoint. This **Event Frame Duration** will be the **target variable** of our predictions. Notice the Event Frame attributes on the left of your Pi Vision screen. Those are the attributes that we have included in our Event Frame and will be using some of those as our **features** (also called **predictor variables**) in order to predict the duration of the startup events.

Create Events Table	Edit Search Criteria
Attributes	
VAVCO startup - VAVCO 5-10 - 2018-10-01 03:...	
End Time Values	
Room Temperature when setpoint reached: ...	
Setpoint Offset at end time: 0.5	
Setpoint reached: True	
Setpoint when setpoint reached: 73 °F	
Start Time Values	
% Cooling at VAV Start: 56.9	
Actual Airflow at VAV Start: 34	
Damper Position at VAV Start: 0.66667 %	
Element Name: VAVCO 5-10	

2.2 Explore generated Event Frames

Let's have a closer look into our Event Frames to see what information exactly we have included in those.

1. Open up PI System Explorer and select the Event Frames pane.



2. Expand the “VAVCO startups” event Frame search and select the first Event Frame from the list and choose the “Attributes: tab. You should be able now to see the available attributes of this event frame:

Event Frames

VAVCO startups

VAVCO startup - VAVCO 3-09 - 2018-06-06 07:01:23.880

General

Child Event Frames

Referenced Elements

Attributes

Filter

Name

Value

Category: End Time Values

Room Temperature when setpoint reached

73.5 °F

Setpoint Offset at end time

0.5

Setpoint reached

True

Setpoint when setpoint reached

73 °F

Category: Start Time Values

% Cooling at VAV Start

40.3749694824219 %

Actual Airflow at VAV Start

0 ft3/min

Damper Position at VAV Start

50 %

Element Name

VAVCO 3-09

Outside Air Temperature at VAV Start

53.4508056640625 °F

Outside Relative Humidity at VAV Start

83.8206024169922 %

Room Temperature at VAV Start

75.5 °F

Setpoint at VAV Start

73 °F

Setpoint Offset at start time

2.5

Space Humidity at VAV Start

39.5 %

There are two categories of attributes labeled End Time Values and Start Time values to indicate values that were captured at the End and at the Start of the Event Frame respectively. Below you will find a short description of the attributes that we will mostly focus on:

- **% Cooling at VAV start** -> The “cooling rate” at the start of the event
- **Actual Airflow at VAV Start** -> The airflow at the start of the event
- **Damper Position at VAV Start** -> The position of the damper at the start of the event
- **Outside Air Temperature at VAV Start** -> The temperature outside the building at the start of the event which is measured by a weather station placed at the top of the building

- **Outside Relative Humidity at VAV Start** -> The outside Relative Humidity at the start of the event
 - **Room Temperature at VAV Start** -> The room temperature at the start of the event
 - **Setpoint at VAV Start** -> The occupied setpoint at the start of the event. We have captured this attribute because the setpoint can be changed manually and it's not always constant.
 - **Setpoint Offset at start time** -> Calculated attribute to calculate how many degrees off the setpoint we are at the start of the event
 - **Space Humidity at VAV Start** -> The humidity of the room at the start of the event
 - **Setpoint reached** -> indicator attribute to indicate if the setpoint was actually reached within the period of the event frame or not. This is because during some very hot days, the setpoint is actually never reached within the course of a day. Since we are only interested in the daily startup events, we have a condition in our event frame generation to close the events at 8 AM, even if the setpoint hasn't been reached, and we need this attribute as a filter to filter out those events later from our analysis.
3. At this point it's worth mentioning that we will not go deeper in the Event Frame generation process, as we assume that you are already familiar with this, and the purpose of this lab is actually an introduction to Data Science. However, if you want to have a look at how are the startup Events captured, you can go to *Library > Element Templates > VAVCO > Analysis Templates > Startup Event* and look at the start and end triggers of the Event Frame generation analysis that have been defined.

[Create a new notification rule template for Start](#)

Example Element: [Select an example element](#)

Event Frame Template: VAVCO startup

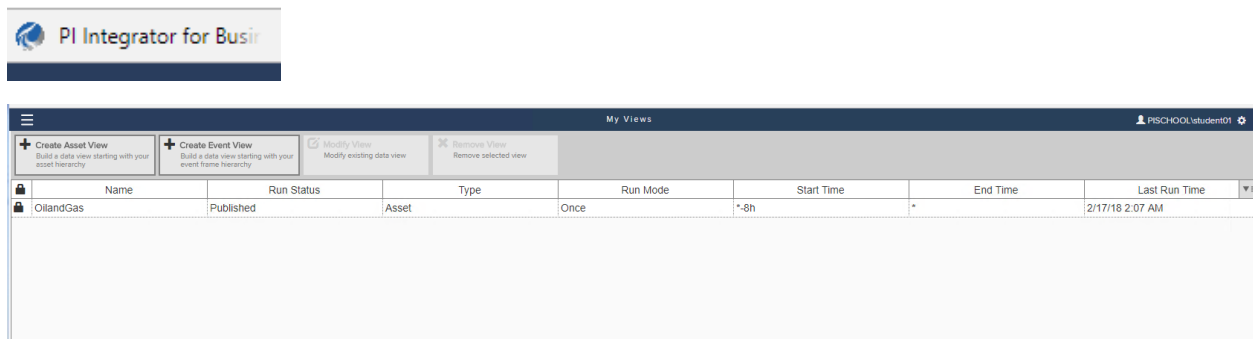
Name	Expression	True for	Severity	Value at Evaluation	Value at Last Trigg
Start triggers					
StartTrigger1	('% cooling'>1) and 'Room Temperature'-'Occupi	Set (optional)	None		
	<pre>('% cooling'>1) and 'Room Temperature'-'Occupied Setpoint' > 0.5 and Hour('*')<=7</pre>				
End trigger					
EndTrigger	(Abs('Room Temperature'-'Occupied Setpoint'))				

Part 3 – Exploratory Data Analysis

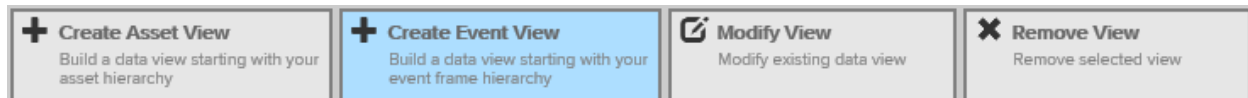
3.1 Publish dataset using the PI Integrator for Business Analytics

(Optional – Demonstration)

1. Access the PI Integrator for BA by opening Google Chrome and selecting its bookmark (<https://pisrv01:1001/>) and enter the credentials provided to log into your VM. You should see the main page of the integrator, showing a list of previously configured views.



2. To get started, select **Create Event View** from the PI Integrator’s top menu.



3. Next, you will be prompted to give your Event View a name, **“VAVCO startups”**. Click **Create View**.

4. Click **“Create a New Shape”**

The first step in using the PI Integrator for BA is to select the AF Server and Database contacting data you want to publish.

- From the dropdown menu, select the AF **Server PISRV01** and the AF **Database Lab Building Data**.

The Event Frames for this model will be listed below. Selecting an Event Frame will show the list of Attributes included in the Event Frame record as determined by its template defined in AF.

- Select and drag the Event Frame at the top of the list and drop it into the middle area under Event Shape.

Select Data > Modify View > Publish

Source Events

Server: PISRV01

Database: Lab Building Data

Enter Event name or string match pattern

Event Frames Assets

- VAVCO startup - VAVCO 2-03 - 2018-06-06 07:01:25.208
- VAVCO startup - VAVCO 2-03 - 2018-06-07 07:01:37.595
- VAVCO startup - VAVCO 2-03 - 2018-06-08 07:02:27.965
- VAVCO startup - VAVCO 2-03 - 2018-06-11 07:03:04.402
- VAVCO startup - VAVCO 2-03 - 2018-06-12 07:03:17.721
- VAVCO startup - VAVCO 2-03 - 2018-06-13 07:00:41.590
- VAVCO startup - VAVCO 2-03 - 2018-06-14 07:00:51.431
- VAVCO startup - VAVCO 2-03 - 2018-06-15 07:01:01.978
- VAVCO startup - VAVCO 2-03 - 2018-06-18 07:01:29.008
- VAVCO startup - VAVCO 2-03 - 2018-06-19 07:01:39.929

- Now, let's add the attributes into our event shape. To do that, just do a "Select All" and drag and drop the attributes in the "Event Shape".

VAVCO startup - VAVCO 2-03 - 2017-07-07 02:57:52.013

Show More

Attributes Filter

Deselect All


- % Cooling at VAV Start
- Actual Airflow at VAV Start
- Damper Position at VAV Start
- Element Name
- Outside Air Temperature at VAV Start
- Outside Relative Humidity at VAV Start
- Room Temperature at VAV Start

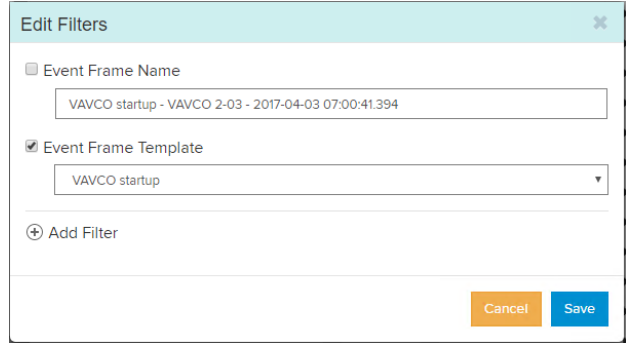
Search Shape

Event Shape

- VAVCO startup
- % Cooling at VAV Start
- Actual Airflow at VAV Start
- Damper Position at VAV Start
- Element Name
- Outside Air Temperature at VAV Start
- Outside Relative Humidity at VAV Start
- Room Temperature at VAV Start
- Room Temperature when setpoint reached
- Setpoint Offset at end time
- Setpoint Offset at start time
- Setpoint at VAV Start
- Setpoint reached
- Setpoint when setpoint reached
- Space Humidity at VAV Start
- Zone Size

So far, we have found one Event Frame match. We need to get all of them.

- Click on the editing pencil  next to the Event Frame shown in the **Event Shape** section. In the **Edit filters** dialog, uncheck the **Event Frame Name** and check the **Event Frame Template** to include all Event Frames derived from the **VAVCO startup** template.

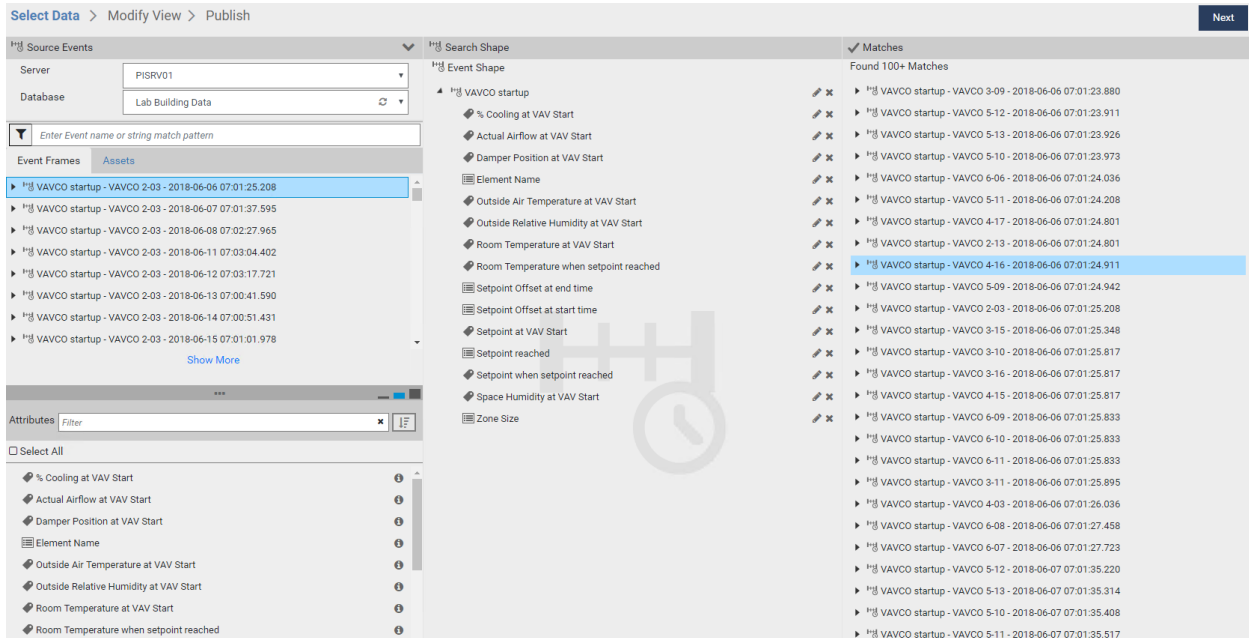


The 'Edit Filters' dialog box has a title bar with a close button. It contains two sections: 'Event Frame Name' with an unchecked checkbox and a text input field containing 'VAVCO startup - VAVCO 2-03 - 2017-04-03 07:00:41.394'; and 'Event Frame Template' with a checked checkbox and a dropdown menu showing 'VAVCO startup'. Below these is an 'Add Filter' button with a plus icon. At the bottom right are 'Cancel' and 'Save' buttons.

- Click **Save**.

You should now see 100+ matches in the Matches section of the Event view. Those are all the event frames that have been generated by the **VAVCO startup** even frame template.

Here is what you browser should look like:



The screenshot shows the BA user interface with the 'Event View' selected. The top navigation bar includes 'Select Data', 'Modify View', 'Publish', and a 'Next' button. The main area is divided into three panels: 'Source Events', 'Search Shape', and 'Matches'. The 'Source Events' panel shows a list of event frames, with 'VAVCO startup - VAVCO 2-03 - 2018-06-06 07:01:25.208' selected. The 'Search Shape' panel shows a list of event shapes, with 'VAVCO startup' selected. The 'Matches' panel shows a list of matches, with 'VAVCO startup - VAVCO 4-16 - 2018-06-06 07:01:24.911' selected. The 'Matches' panel also displays a large watermark '100+' and a 'Found 100+ Matches' message.

- Click **Next** at the top right-hand side of the page to move to the **Modify View** page.

Next, we need to add a couple of Time Columns to the **Event View** as well as remove some that are not required.

- From the menu at the top of the PI Integrator for BA user interface, click **Add Columns**.

12. Select the **Time Column** tab.

Add Column

Data Column

Time Column

Static Value

Select Time Column Options for Local

TimeStamp(2/21/2019 5:45:13 AM)

Year(2019)

Month Name(February)

Week of the Year(8)

Day(21)

Hour(5)

Minute(49)

Second(49)

Milliseconds(190)

UTC Seconds(1550756989.19)

UTC Milliseconds(1550756989190)

Ticks(636863537891900000)

Time Zone Offset(480)

Event Frame Start Time(Event Frame Start Time (Local))

Event Frame End Time(Event Frame End Time (Local))

Month(Local)

Day of the Week(Local)

Event Frame Duration(Event Frame Duration)

Cancel

Display 5 time columns

13. Select the **TimeStamp (Local)** column from the right hand section and remove it from the list. Also select the **Month** and **Day of the Week** columns from the left hand section and add them to the list. Select “**Display 5 time columns**” to confirm your selection.

14. Back in the main panel, select the newly added **Event Frame Duration Hour** column and change the **Data Content** from **Hour** to **Minute**. Rename the column to **Event Framer Duration Minute** and click **Apply Changes**. This is done because we are going to look into predicting the Duration up to the minute level.

		Start Time	End Time	
		6/6/18 7:01 AM	*	
Event Frame Duration Minute	% Cooling at VAV Start	Actual Airflow at VAV Start	Damper Position at VAV Start	
55	40.375	0	50	
40	33.5	0	50	
35	26.642	38	50.667	
60	33.067	0	7.333	
25	33.717	291	50.667	
35	33.467	0	50	
25	19.9	0	50	
370	96.55	56	63.889	
100	26.75	49	0	
30	26.725	0	50	
40	26.708	182	58.333	
50	40.3	65	50	
35	33.425	0	50	

Name

Event Frame Duration Minute

Reset Name to Default

Data Content

Minute

Time Context

Event Frame Duration

Data Type

Integer

Remove Column

Apply Changes

We are now ready to publish the **Event View**.

15. Select **Apply** in the upper right-hand corner of the page and **Next** to move to the **Publish** page of the PI Integrator for BA.
16. In the Target Configuration change from **PI View** to **CSV**.
17. Click on **Publish** to get the PI Integrator to start publishing the dataset to a file. You must **Confirm** that this is ok. You should be directed back to the page showing the existing list of views. The bottom of the page shows a run status of the publish action.

Target Configuration

CSV ▼

Run Mode

- ☒ Run Once
- ☐ Run on a Schedule

Summary

Shape and Matches

- There are 100+ Matching Instances

Timeframe and Interval

- Your Start Time is 6/6/18 12:00:00 AM
- Your End Time is *
- Your Time Interval gets an interpolated measurement Every 1 minute

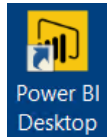
Publish

3.2 Data Exploration using Microsoft Power BI Desktop

3.2.1 Loading the Data in Power BI

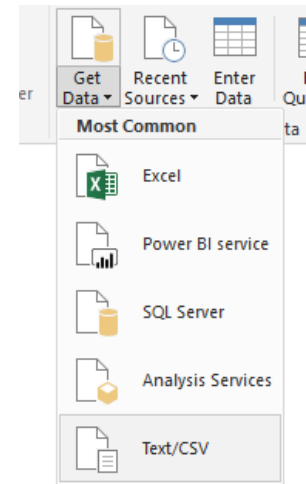
Now we need to load the dataset that we've published, using the PI integrator for Business Analytics, in Power BI.

1. Open Microsoft Power BI Desktop using the shortcut on the VM desktop.

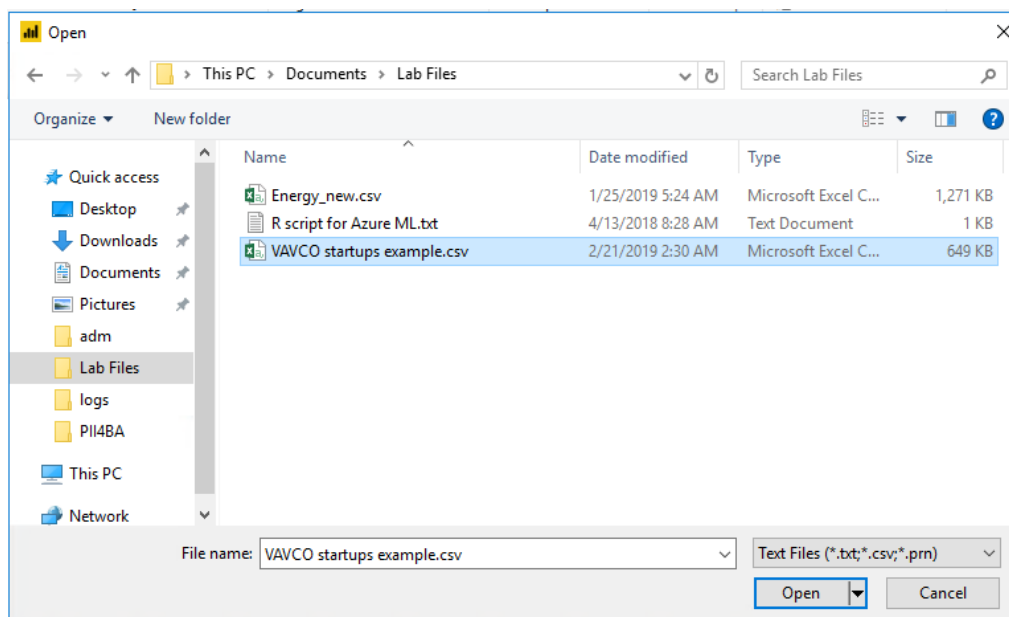


2. From the menu ribbon, click on **Get Data** to expose the dropdown (right). Select **Text/CSV**. This will open the **Get Data** dialog (below).

As our input dataset, we are simply going to use a pre-exported file and not the one that we just generated from the Integrator, just to make sure that we all start from the same base in case there were any issues or discrepancies during exporting the data from the integrator.



3. Navigate to **Documents\Lab Files** and choose **VAVCO startups example.csv** and then **Open**.



4. Once connected, the first rows of the dataset will be shown in preview mode. Click **Load** to bring the *VAVCO startup example* dataset into your report.

VAVCO startups example.csv

File Origin: 65001: Unicode (UTF-8) | Delimiter: Comma | Data Type Detection: Based on first 200 rows

Id	VAVCO startup	Event Frame Start Time (Local) TimeStamp	Event Frame End Time (Local) TimeStamp	Ev
1	VAVCO startup - VAVCO 3-09 - 2018-06-06 07:01:23.880	6/6/2018 7:01:24 AM	6/6/2018 7:56:25 AM	
2	VAVCO startup - VAVCO 5-12 - 2018-06-06 07:01:23.911	6/6/2018 7:01:24 AM	6/6/2018 7:41:24 AM	
3	VAVCO startup - VAVCO 5-13 - 2018-06-06 07:01:23.926	6/6/2018 7:01:24 AM	6/6/2018 7:36:25 AM	
4	VAVCO startup - VAVCO 5-10 - 2018-06-06 07:01:23.973	6/6/2018 7:01:24 AM	6/6/2018 8:01:24 AM	
5	VAVCO startup - VAVCO 6-06 - 2018-06-06 07:01:24.036	6/6/2018 7:01:24 AM	6/6/2018 7:26:25 AM	
6	VAVCO startup - VAVCO 5-11 - 2018-06-06 07:01:24.208	6/6/2018 7:01:24 AM	6/6/2018 7:36:25 AM	
7	VAVCO startup - VAVCO 4-17 - 2018-06-06 07:01:24.801	6/6/2018 7:01:25 AM	6/6/2018 7:26:27 AM	
8	VAVCO startup - VAVCO 2-13 - 2018-06-06 07:01:24.801	6/6/2018 7:01:25 AM	6/6/2018 1:11:29 PM	
9	VAVCO startup - VAVCO 4-16 - 2018-06-06 07:01:24.911	6/6/2018 7:01:25 AM	6/6/2018 8:41:25 AM	
10	VAVCO startup - VAVCO 5-09 - 2018-06-06 07:01:24.942	6/6/2018 7:01:25 AM	6/6/2018 7:31:26 AM	
11	VAVCO startup - VAVCO 2-03 - 2018-06-06 07:01:25.208	6/6/2018 7:01:25 AM	6/6/2018 7:41:26 AM	
12	VAVCO startup - VAVCO 3-15 - 2018-06-06 07:01:25.348	6/6/2018 7:01:25 AM	6/6/2018 7:51:27 AM	
13	VAVCO startup - VAVCO 3-10 - 2018-06-06 07:01:25.817	6/6/2018 7:01:26 AM	6/6/2018 7:36:26 AM	
14	VAVCO startup - VAVCO 3-16 - 2018-06-06 07:01:25.817	6/6/2018 7:01:26 AM	6/6/2018 8:36:27 AM	
15	VAVCO startup - VAVCO 4-15 - 2018-06-06 07:01:25.817	6/6/2018 7:01:26 AM	6/6/2018 7:31:26 AM	
16	VAVCO startup - VAVCO 6-09 - 2018-06-06 07:01:25.833	6/6/2018 7:01:26 AM	6/6/2018 7:51:27 AM	
17	VAVCO startup - VAVCO 6-10 - 2018-06-06 07:01:25.833	6/6/2018 7:01:26 AM	6/6/2018 8:06:28 AM	
18	VAVCO startup - VAVCO 6-11 - 2018-06-06 07:01:25.833	6/6/2018 7:01:26 AM	6/6/2018 7:46:26 AM	
19	VAVCO startup - VAVCO 3-11 - 2018-06-06 07:01:25.895	6/6/2018 7:01:26 AM	6/6/2018 8:01:27 AM	
20	VAVCO startup - VAVCO 4-03 - 2018-06-06 07:01:26.036	6/6/2018 7:01:26 AM	6/6/2018 8:46:26 AM	

Load Edit Cancel

5. Once the data is loaded, look at the upper left-hand edge of the Power BI Desktop window. You will see three gray icons. These icons, from top to bottom, are used to move between the *Report*, *Data* and *Relationship* views within Power BI Desktop. Selecting the **Data** icon will expose a view of the *Product Inventories* data loaded into Power BI Desktop. The column headers are identical to the ones selected during the shaping step when using the PI Integrator for BA. This data is stored in the memory of your VM to enable Power BI to respond quickly to your analysis.



6. Navigate back to the *Report* view to access the blank reporting canvas. The following steps describe how to configure each chart in the example report. Please feel free to arrange and resize them any way you like.

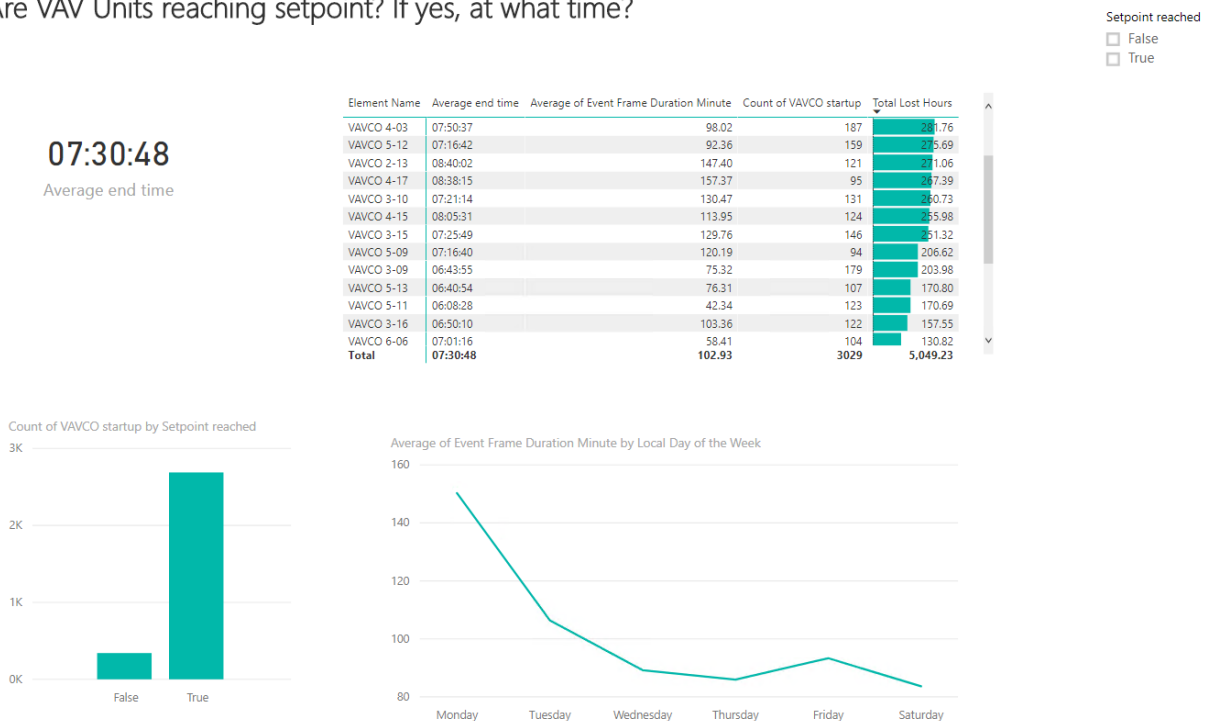
3.2.2 Creating Reports for Data Exploration

In this section we are going to create some reports in order to explore our data and find relationships that will help us on building our predictive model, later on in the lab.

First of all, we want to create a report that will look into the average time that the VAVCO units actually reach the setpoint. This will be very important for our project, in order to verify that the scope of the project is valid and there is room for improvement in the startup process of the units. If the majority of the VAVCO units is already reaching the setpoint at 7 AM then there is no need to try and optimize this further, but if not we will have a measure to quantify the value of the project. So let's get into it:

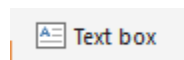
Report Title – “Are VAV Units reaching setpoint? If yes, at what time?”

Are VAV Units reaching setpoint? If yes, at what time?



In the above screenshot you can see a preview of the final report that we are going to build. Let's now start building the pieces of the report.

1. From the **Home** tab on the menu ribbon, click **Text box** to insert one on the report canvas.
2. Select a larger text font and give your report a title, like **“Are VAV Units reaching setpoint? If yes, at what time?”**.



Card – Average end time

We want to show the average time that the VAVCO units manage to reach the setpoint. For that, we will need to add a few columns and measurements that are missing from our dataset.

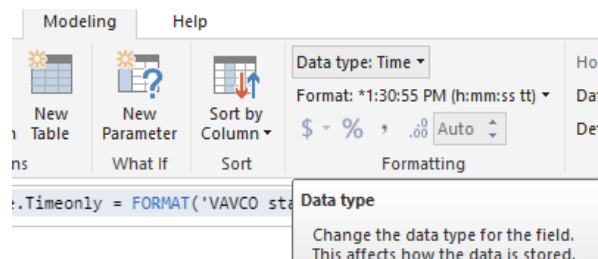
3. Right-click on the table name **VAVCO startups example** to expand the menu shown at right. Select **New column** and two things will happen; a new field, named *Column*, appears in the table's field list and the formula bar is inserted at the top of the Power BI report canvas as shown below.



4. Copy and paste into the formula bar and **Enter** this calculated column to the **VAVCO startups example** table.

EndTime.Timeonly = FORMAT('VAVCO startups example'[Event Frame End Time (Local) Timestamp], "hh:mm:ss")

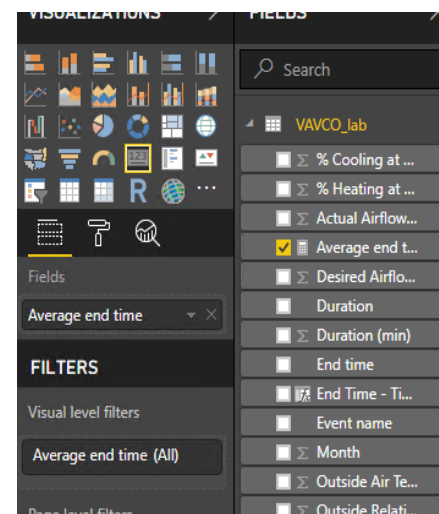
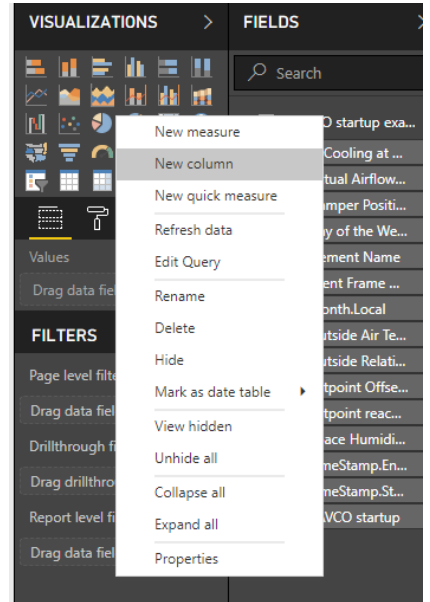
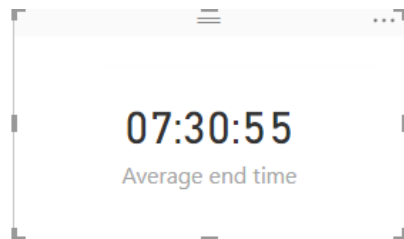
5. Select the **EndTime.Timeonly** column and change it to **Time** format from the Modelling menu on the Power BI menu ribbon.



6. Now Right-click again on the **VAVCO startup example** table and select **New Measure**. Copy and paste into the formula bar and **Enter** this measure to the **VAVCO startup example** table.

Average end time = FORMAT(AVERAGE('VAVCO startups example'[EndTime.Timeonly]), "hh:mm:ss")

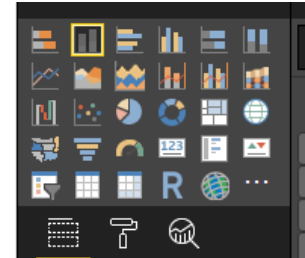
7. Let's add this measure now into our report. Click a blank space on the report canvas. Select a card visual from the pallet in the visualization pane. Add the **Average end time** from the fields list of **VAVCO startup example** into the card



We also want to examine what percentage of startup events actually manages to reach the setpoint by 7 AM in the morning. To answer this question, we are going to use the “**Setpoint reached**” attribute from our dataset.

Stack column chart – Setpoint reached

8. Click a blank space on the report canvas. Select a **stacked column chart** visual from the pallet in the visualization pane.



9. Add the **Setpoint reached** from the fields list of **VAVCO startup example** to the **Axis** field of the stacked column chart and the **VAVCO startup** field in the **Values** field of the column chart.

Your stacked column chart should look similar to the one below:

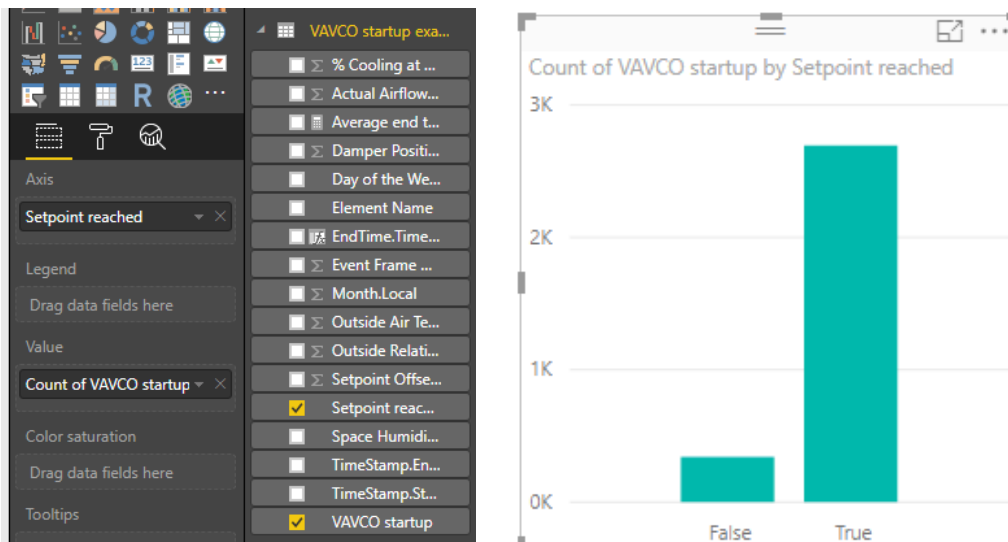
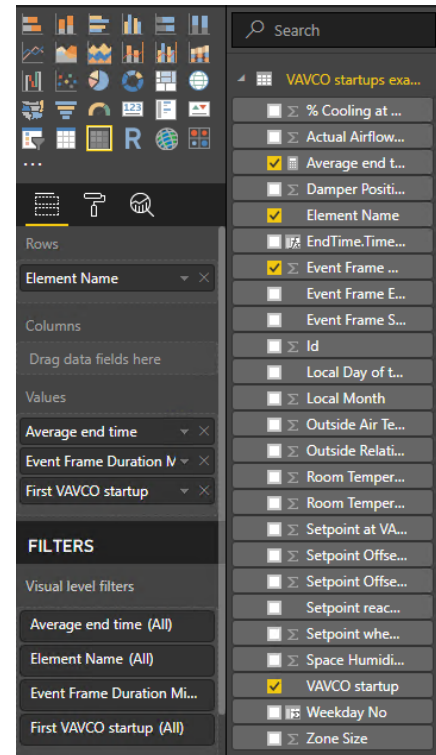


Table of Average end time

10. Click on a blank space on the report canvas and select a **Matrix** visual.
11. Populate the matrix by selecting and dragging each of the fields **Average end Time**, **Event Frame Duration.Minute** and **VAVCO startup** into the **Values** bucket of the table's field well and the **Element Name** into the **Rows** field.

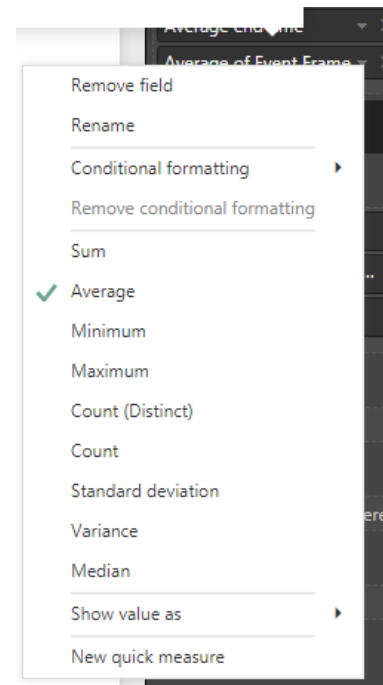
By default, the **Event Frame Duration.Minute** column shows the total (sum) volume contained in each tank for all of 2016. It may be more useful to show the **average Event Frame Duration.Minute** instead of the sum.



12. To change the aggregation from *Sum* to *Average*, return to the **Values** field well and click the small arrow next to the **Event Frame Duration Minute** field.

13. Do the same for the VAVCO startup field which should be currently labeled as **First VAVCO startup**. Click the small arrow next to it and change it to **Count**

The exposed dropdown allows you to modify the table calculation to show the average Event Frame Duration Minute. Charts containing numeric values in Power BI will always default to aggregating by sum, so keep this in mind as you are building your reports.



Element Name	Average end time	Average of Event Frame Duration Minute	Count of VAVCO startup
VAVCO 2-03	06:42:19	39.39	90
VAVCO 2-09	09:35:41	205.49	111
VAVCO 2-11	07:16:26	78.83	77
VAVCO 2-13	08:40:02	147.40	121
VAVCO 3-09	06:43:55	75.32	179
VAVCO 3-10	07:21:14	130.47	131
VAVCO 3-11	07:32:23	131.48	154
VAVCO 3-15	07:25:49	129.76	146
VAVCO 3-16	06:50:10	103.36	122
VAVCO 4-03	07:50:37	98.02	187
VAVCO 4-15	08:05:31	113.95	124
VAVCO 4-16	10:10:37	249.06	112
VAVCO 4-17	08:38:15	157.37	95
Total	07:30:48	102.93	3029

We now have a table of the Average time to reach the setpoint, Average of how long it takes to reach the setpoint and Count of startup events in our dataset, by VAV unit.

Add Lost Hours

From the matrix that we created, we can notice that there is a variation on the average time when the different units reach the setpoint. As a next step, it would be nice to have a column showing the **Total Lost hours** as a metric of each unit's performance in terms of how long before or after 7 AM it reaches the setpoint.

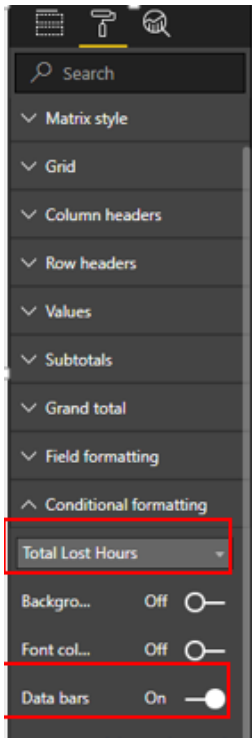
14. Right-click on the table name **VAVCO startups example**, select **New column** and add the following text in the formula bar:

Total Lost Hours = abs("7:00"-VAVCO startups example'[EndTime.Timeonly])*24

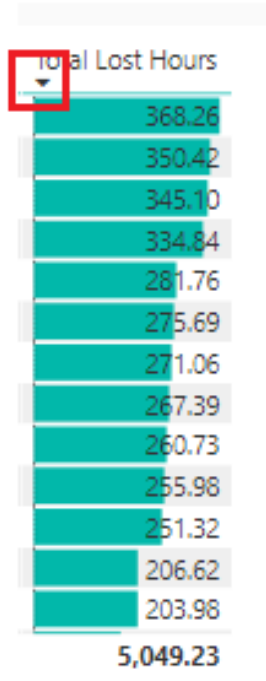
15. Add this new column of **Total Lost Hours** to the table of the VAVCO units leaving the default summary to **Sum**. We can also add some color conditional formatting to this column to make it more visible. To access formatting options for any visual, click on the paint roller icon just below the visual objects pallet.



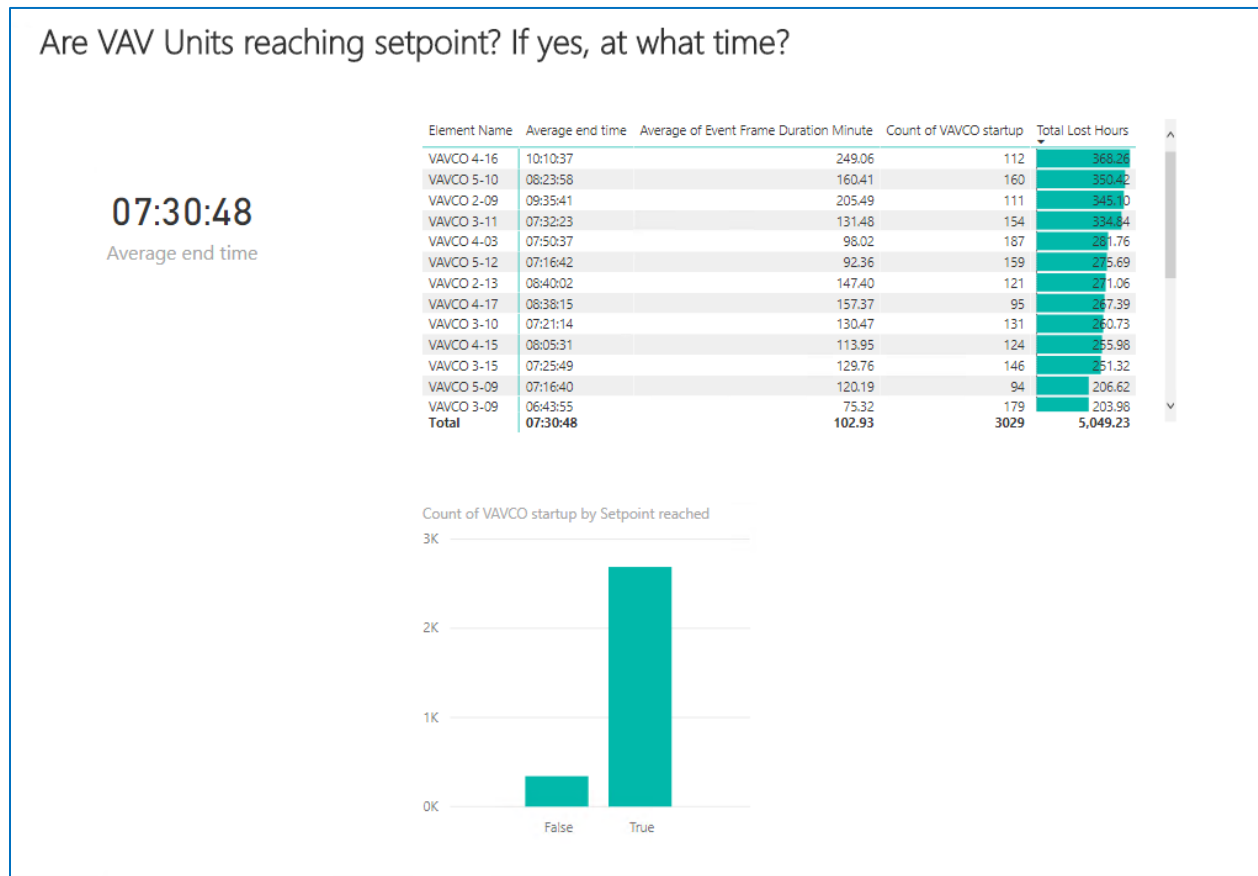
- Expand the **Conditional Formatting** section of the list, select the **Total Lost Hours** column from the dropdown list and enable the **Data Bars** option, as shown in the screenshot.



- You can also sort the rows of the table in descending order, based on the value of their **Total Lost Hours** attribute, by selecting the small arrow below the name of the column.



Below, you can see a screenshot of how your report should look like so far:



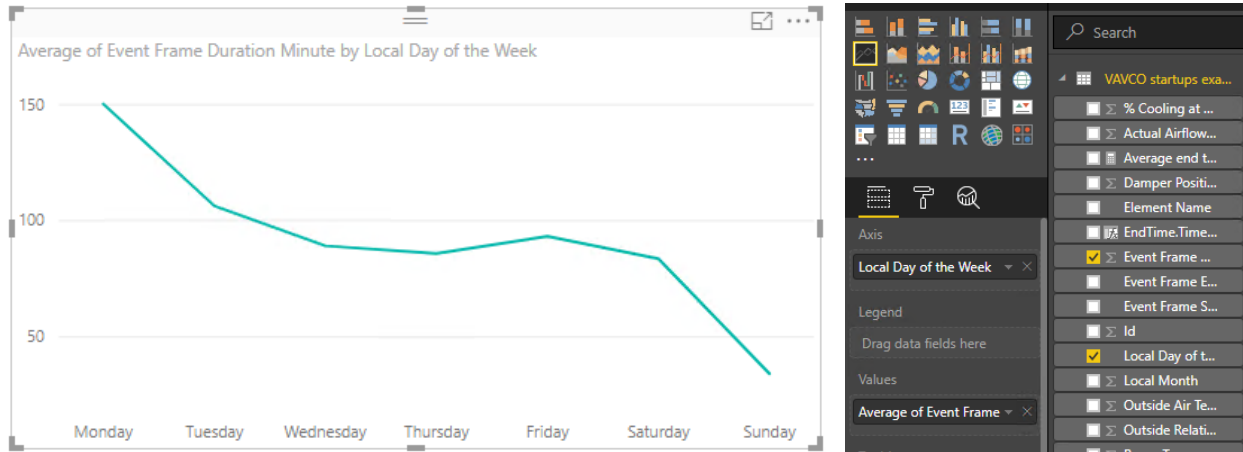
Discussion

We now have some information regarding the overall average time when our units reach the setpoint temperature, as well as how long it takes on average to reach the setpoint and how many Hours are lost because of reaching the setpoint too early or too late. We can see that although on average we reach the setpoint at about 7:30 AM in the morning, there is a lot of variation if we break down the same average by the individual VAVCO units, with some units performing a bit better and some worse. Since our goal is to optimize the startup process in order to reach the setpoint as close as possible to 7 AM, we can conclude that there is room for improvement by further optimizing the units' startup. Specifically, we have wasted more than 5,000 hours of operation within the last 1 year.

Let's add one more visual into our report, just to look at the data by Day of the Week.

Line chart

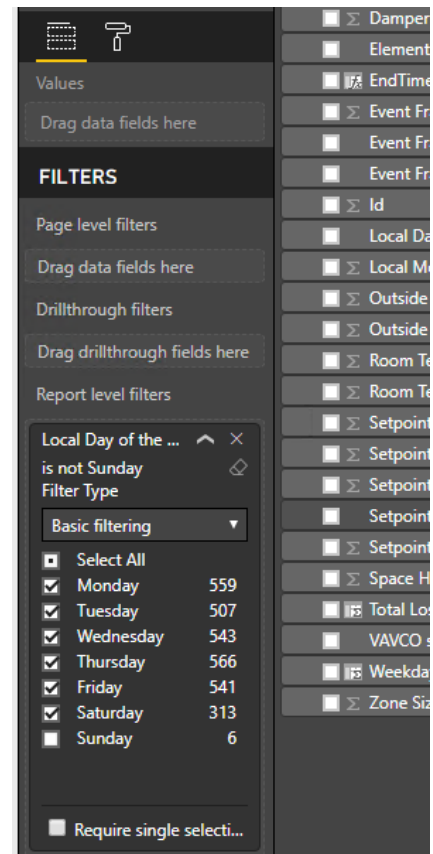
18. Click on a blank space on the report canvas and select the **Line chart** visual.
19. Fill in the fields of the visual with the respective fields as shown in the screenshot, with **Local Day of the Week** in the **Axis** field and **Average of Event Frame Duration** in the **Values** field.



(Hint: Your days will probably not be in order. Ask your instructor how to sort them, or try it yourself! You could also take a peak in the provided **Data Exploration report example.pbix** file, in in your Lab files folder and try to figure out how it's done)

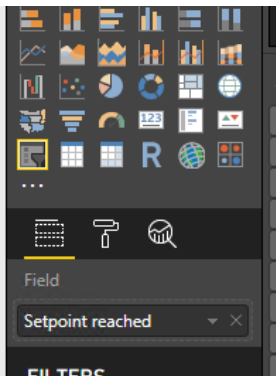
The first thing we notice, is that it looks like we have some startup events on Sunday. That can't be right, since our Subject Matter Expert has confirmed that on Sundays the VAV units are shutdown. This looks like a Data Quality issue, so what we'll do is that we are going to remove this data from our analysis.

20. In order to do that we need to add **Local Day of the Week** under the **Report Level Filters** and unselect Sunday:

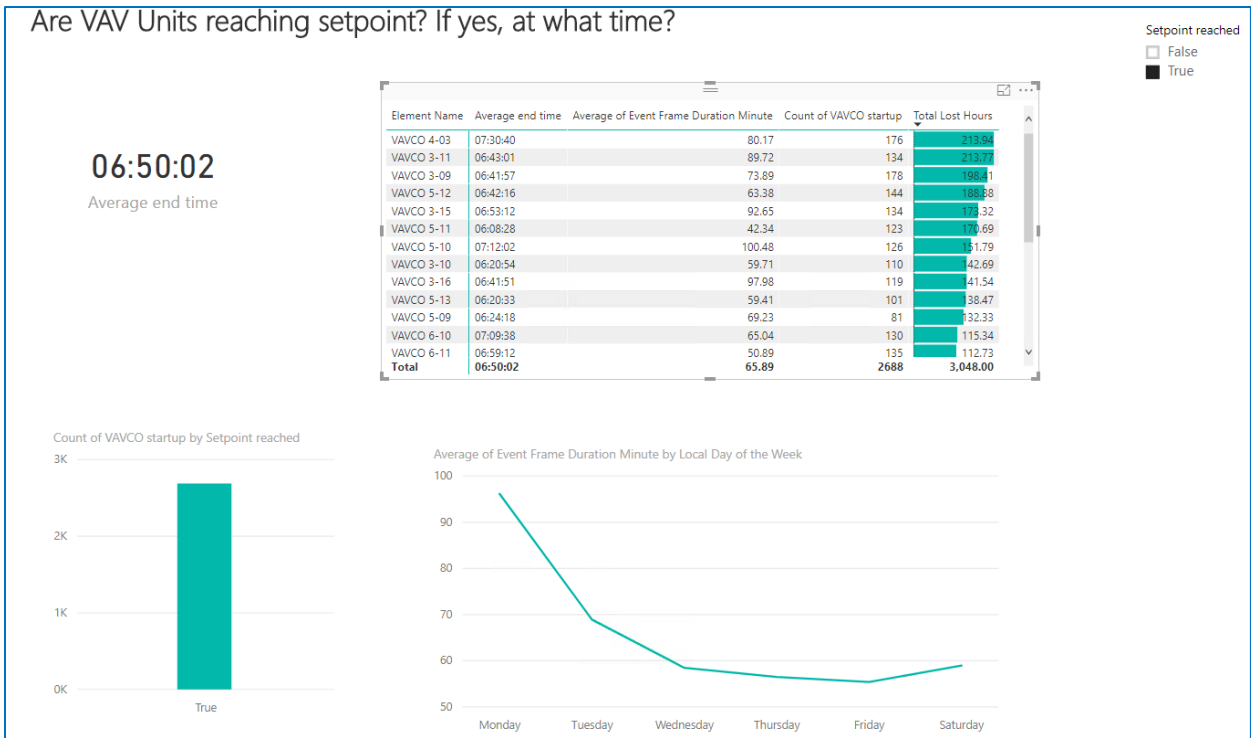


Finally, just because we are mostly interested in the cases when the setpoint was actually reached, we are going to add a filter.

- 21. Click on a blank space on the report canvas, select the *Slicer* visual and add the **Setpoint reached** column to it.



This is what your final report should look like now:





Discussion

The last visuals that we added, gave us some information that could be useful later in our Analysis and building of our predictive model. The **Total Lost Hours** column gives us a clear indication of which are the worst performing units (which in this case is VAVCO 4-16), so that we can focus on them first. Furthermore (if you filter the report only for **Setpoint Reached = True**), the graph of the **Average of Event Frame Duration** by **Local Day of the Week** clearly shows that on Monday the average duration is significantly higher than the rest of the days. This can be explained by the fact that during the weekend, the cooling units are shutdown, so on Monday they have to work harder to bring the temperature back to the desired setpoint.

Are VAV Units reaching setpoint? If yes, at what time?

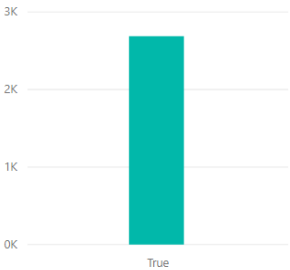
06:50:02

Average end time

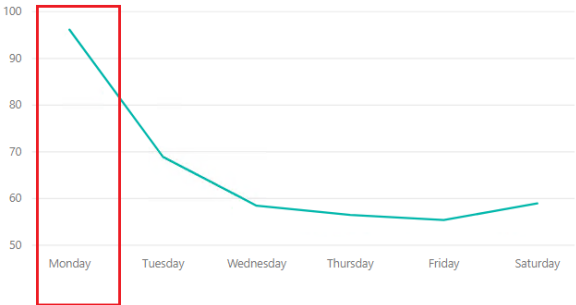
Element Name	Average end time	Average of Event Frame Duration Minute	Count of VAVCO startup	Total Lost Hours
VAVCO 6-10	07:09:38	65.04	130	115.34
VAVCO 6-11	06:59:12	50.89	135	112.73
VAVCO 6-07	07:00:37	49.88	124	111.85
VAVCO 6-09	06:56:30	51.22	131	111.41
VAVCO 4-15	06:43:26	44.32	95	94.34
VAVCO 4-17	06:41:11	57.42	64	91.75
VAVCO 2-11	06:54:32	56.32	72	88.63
VAVCO 6-08	06:41:18	36.61	112	87.57
VAVCO 2-03	06:42:19	39.39	90	83.99
VAVCO 6-06	06:31:17	34.16	95	83.14
VAVCO 2-09	07:15:42	84.98	66	74.35
VAVCO 2-13	07:00:21	57.24	87	69.83
VAVCO 4-16	07:44:03	113.69	61	57.22
Total	06:50:02	65.89	2688	3,048.00

Setpoint reached
☐ False
☒ True

Count of VAVCO startup by Setpoint reached

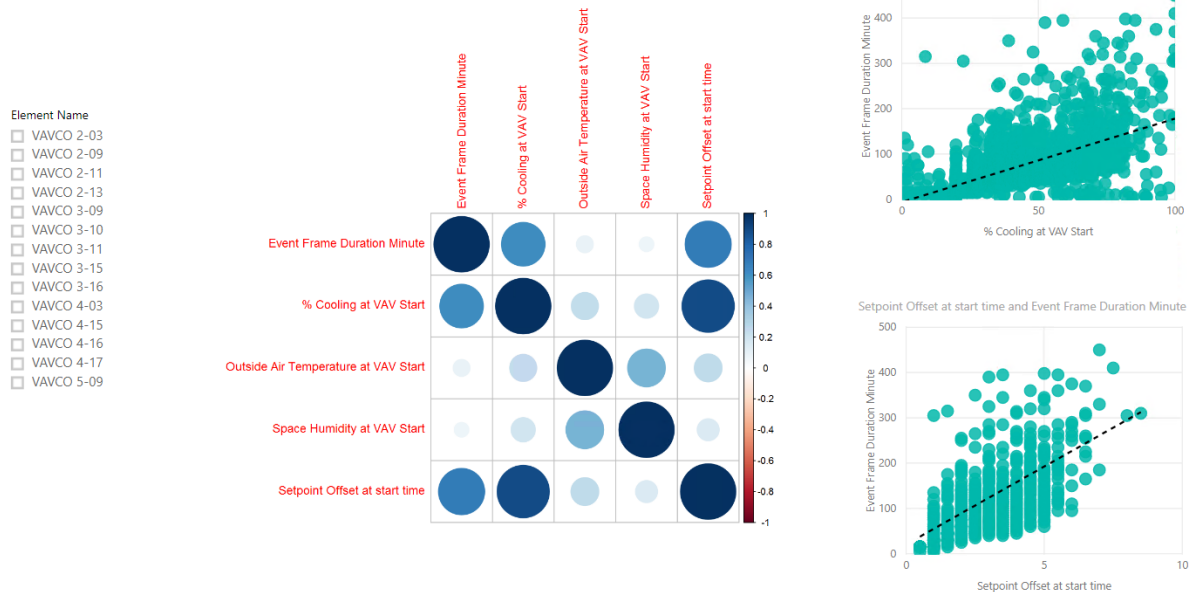


Average of Event Frame Duration Minute by Local Day of the Week



Report Title – “Bivariate Analysis”

Bivariate Analysis



In the above screenshot you can see a preview of the final report that we are going to build. Let’s now start building the pieces of the report.

Since we have determined that there are improvements that can be done for the building startup, we can continue with our Exploratory Data Analysis and try to find relationships in our data that will help us build an accurate predictive model that will predict the amount of time required to reach the setpoint, at any given point of time. Remember that by having this prediction we will be able to decide when is the best time to start the VAVCO units, so that the setpoint is reached as close as possible to 7 AM.

The next step in our Data Analysis will be to create some bivariate plots of the **Duration** (our **target variable**) against the other features of our dataset (**predictor variables**).

At this stage we would go back to our SMEs and discuss with them which variables make most sense to use as predictors. We wouldn’t want to include all the available variables and end up with a model that doesn’t make sense at all in terms of physics or engineering because then we would lose trust. So after our discussion the SMEs said that we should focus on **%Cooling, Outside Temperature, Space Humidity** and **Setpoint Offset**.

Correlation Plot

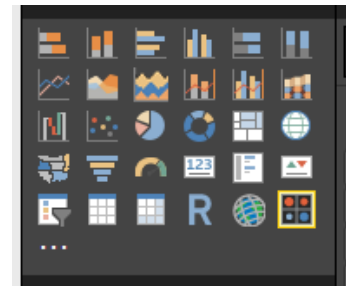
1. Select the **New Page** symbol on the bottom of your report.



Microsoft Power BI is an extensible application. This means that third parties (and you) can develop visuals that can be added to the Power BI environment to make your reports even more interesting. Power BI Custom Visuals can be downloaded from the Microsoft Office Store at, <https://app.powerbi.com/visuals>. We are going to use a custom visual called **Correlation plot** in order to further examine the correlations between our variables.

2. To load the **Correlation plot** into your report, click on the ellipses (...) on the visuals pallet and select **Import from file**. Navigate to **C:\Users\student01.PISCHOOL\Documents\Lab Files\PowerBI-visuals-corrplot.1.0.1.0.pbviz** and **Open** it. You now have another visual to use in your report.

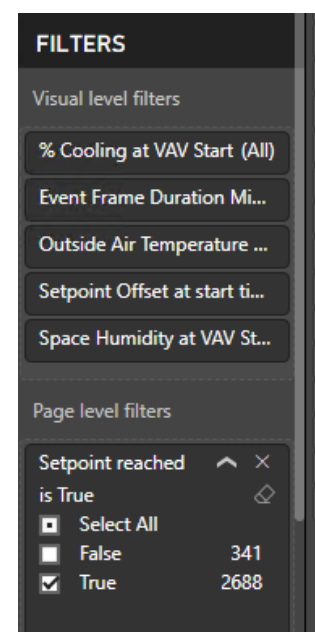
3. Click a blank space in the canvas and select the **Correlation plot** visual that we imported, from the pallet in the visualization pane.

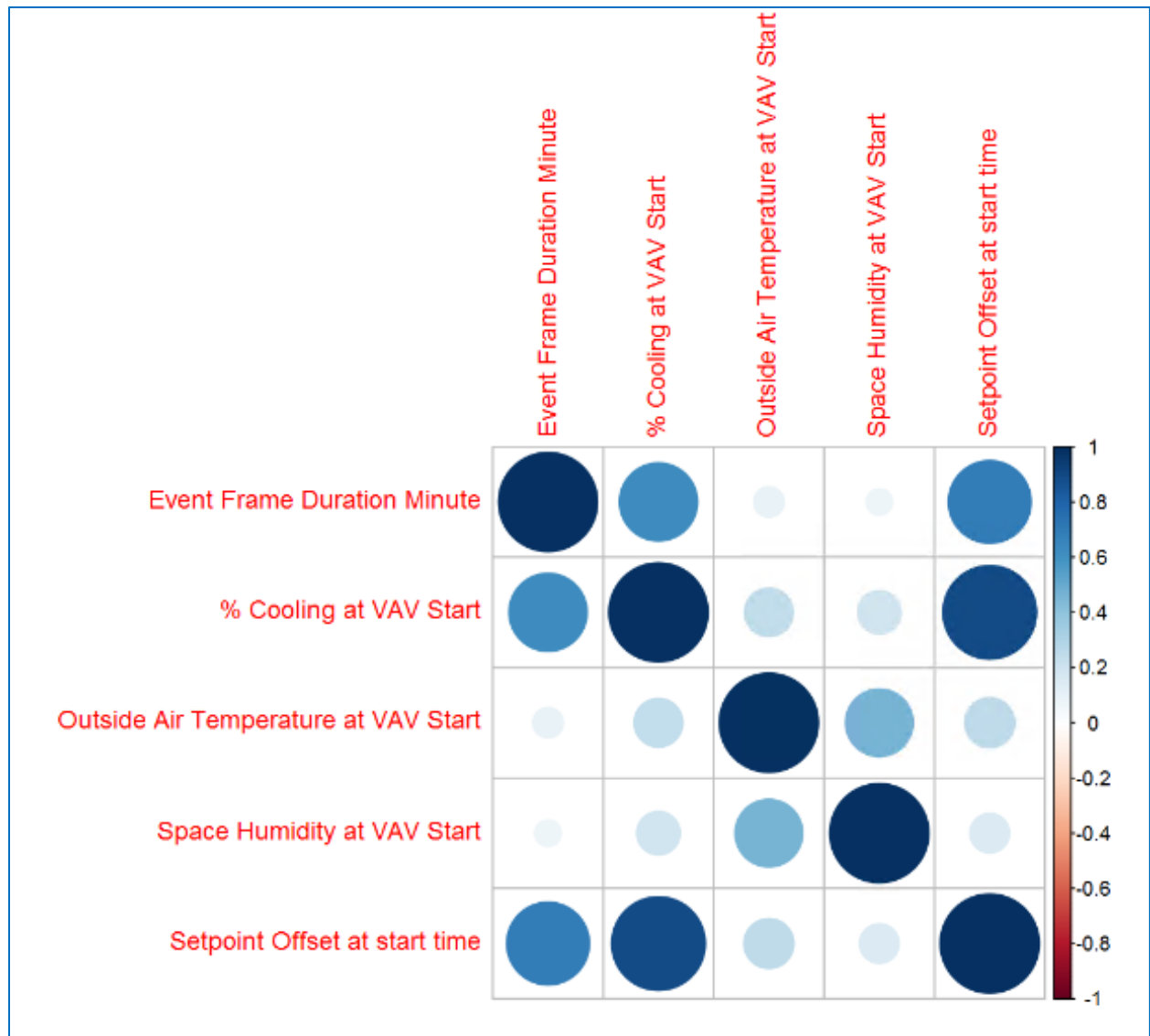


4. Add the following fields in the **Values** bucket of the correlation plot, making sure to change their default summary mode from Sum to Don't Summarize: **Event Frame Duration.Minute**, **% Cooling at VAV start**, **Outside Air Temperature at VAV Start**, **Space humidity at VAV start** and **Setpoint Offset at VAV start**

In this case we are interested to investigate only the startup events when the setpoint was reached, so we can filter out the cases when the setpoint was not reached.

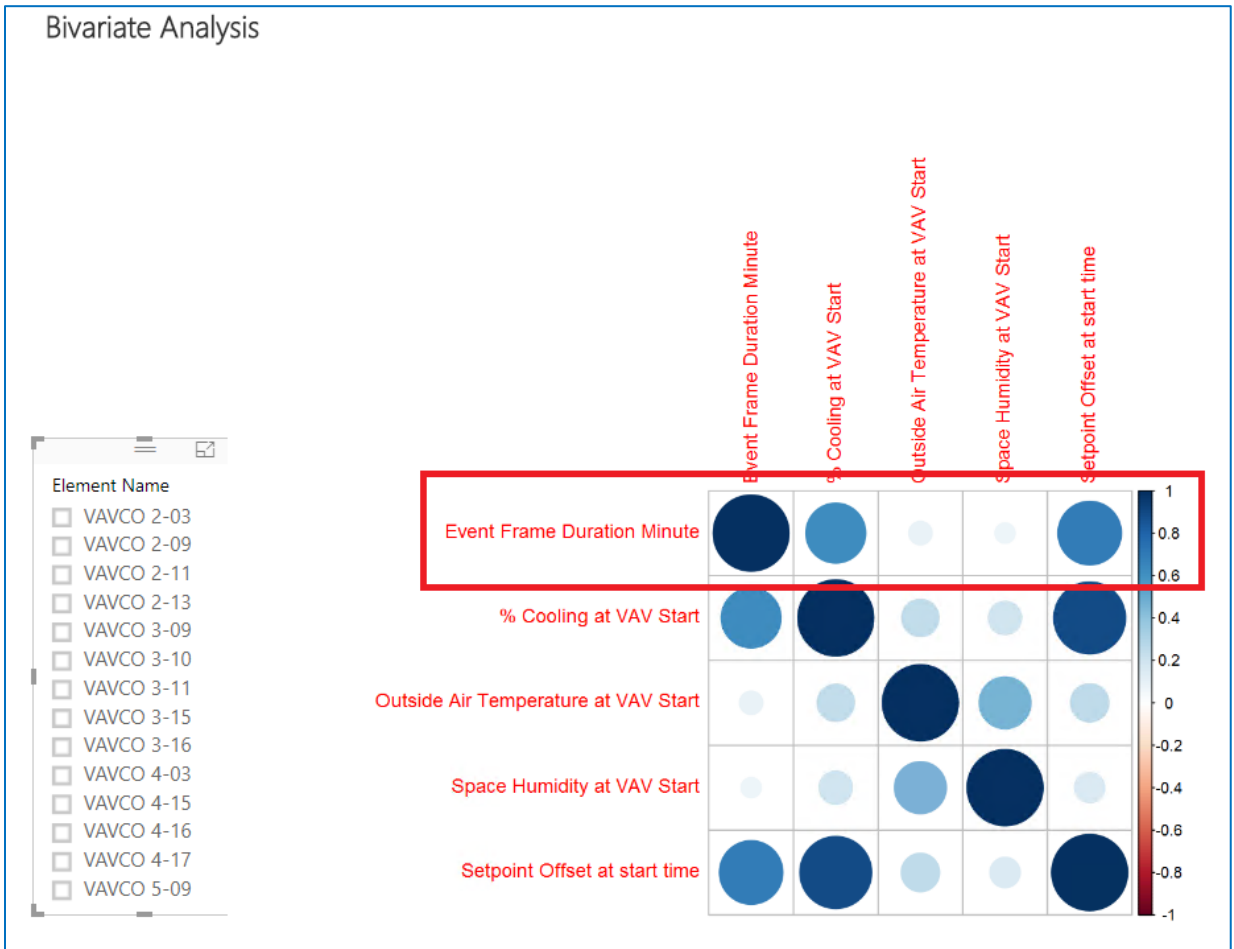
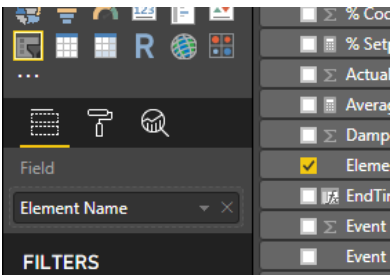
5. To do that, go back to the **Fields** menu, drag the **Setpoint reached** column in the **Page Level Filters** field and select **True**.





Let’s also add a slicer in order to be able to filter our data by VAVCO unit.

- 6. Click a blank space in your canvas and select the **Slicer** visual. Add the **Element Name** column as the slicer’s field.



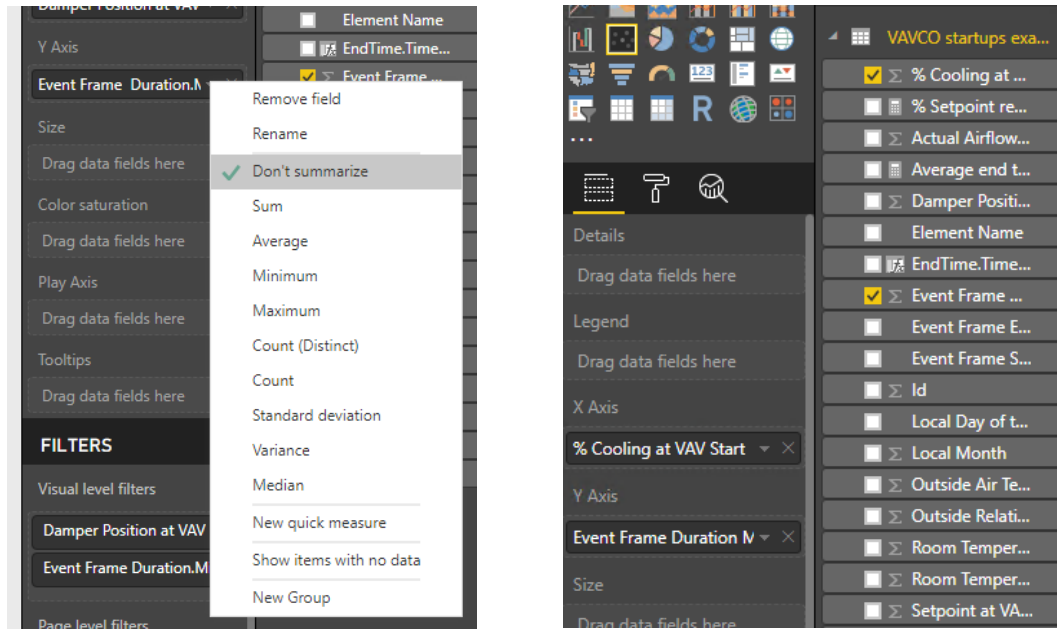
In the correlation plot, the size and color of the circle indicate the degree of correlation (larger, darker circles = stronger correlation); color indicates type of correlation (blue = positive, red = negative)

What we see in the above plot is that the **Event Frame Duration.Minute** attribute, seems to have a strong positive correlation with the **Setpoint Offset at VAV start** and with the **% Cooling at VAV start**.

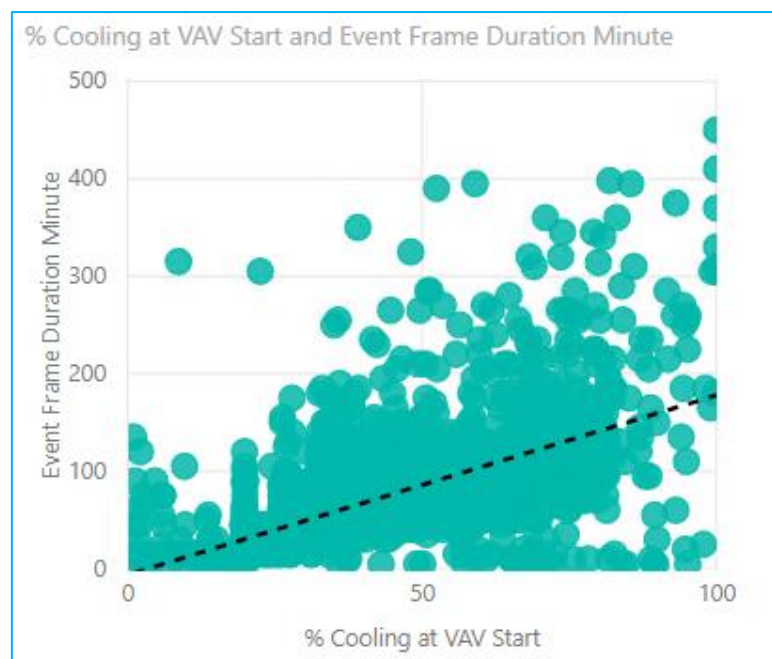
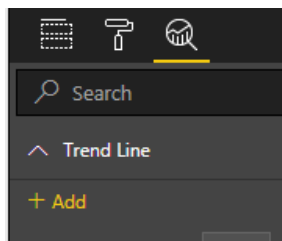
We can confirm that by adding some scatter charts to our report.

Scatter chart

- Click on a blank space in your canvas and select the **scatter chart** symbol from the available symbols library. Add the **% Cooling at VAV start** at the **X axis** field and the **Event Frame Duration Minute** at the **Y axis** field and change their default summary mode from **Sum** to **Don't Summarize**.



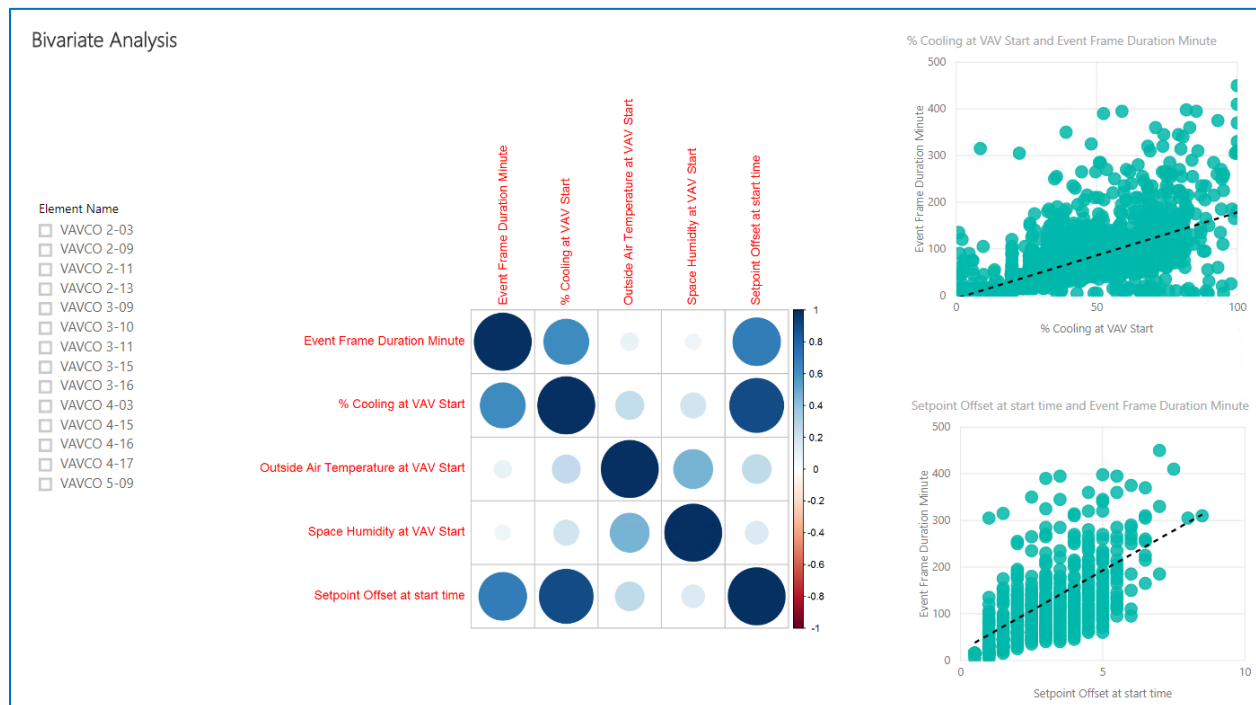
- Next, select the Analytics tab of the scatter chart visual, expand the **Trend Line** section and choose the **Add** option.



We are going to add one more scatter chart, in order to investigate the relationship of the **Event Frame Duration Minute** variable (**target variable**) with the **Setpoint Offset at VAV start** as well.

- Now copy and paste the scatter chart you just created. On the copied scatter chart, change the **X axis** field to **Setpoint Offset at VAV start** (again remember to change the default summary mode from **Sum** to **Don't Summarize**).

This is how your report should look like eventually:



From the scatter charts we can see a linear relationship between the **Event Frame Duration Minute** and the other two features (**Setpoint Offset at VAV start** and **% Cooling at VAV start**) as indicated by our correlation plot. This relationship is more clear if we select a few VAV units from our slicer and look at them individually (pick VAVCO 2-13, VAVCO 4-16, VAVCO 6-08 for example). Notice that the relationship becomes now very obvious.



These graphs gave us very valuable information regarding how to build our predictive model. We have identified that the **Setpoint Offset at VAV start** and **% Cooling at VAV start** are our strongest predictors. **Outside Air Temperature** and **Space Humidity** don't seem to "explain" the **Duration** variable very well.

Part 4 – Modeling and Evaluation

4.1 Building a model in Orange

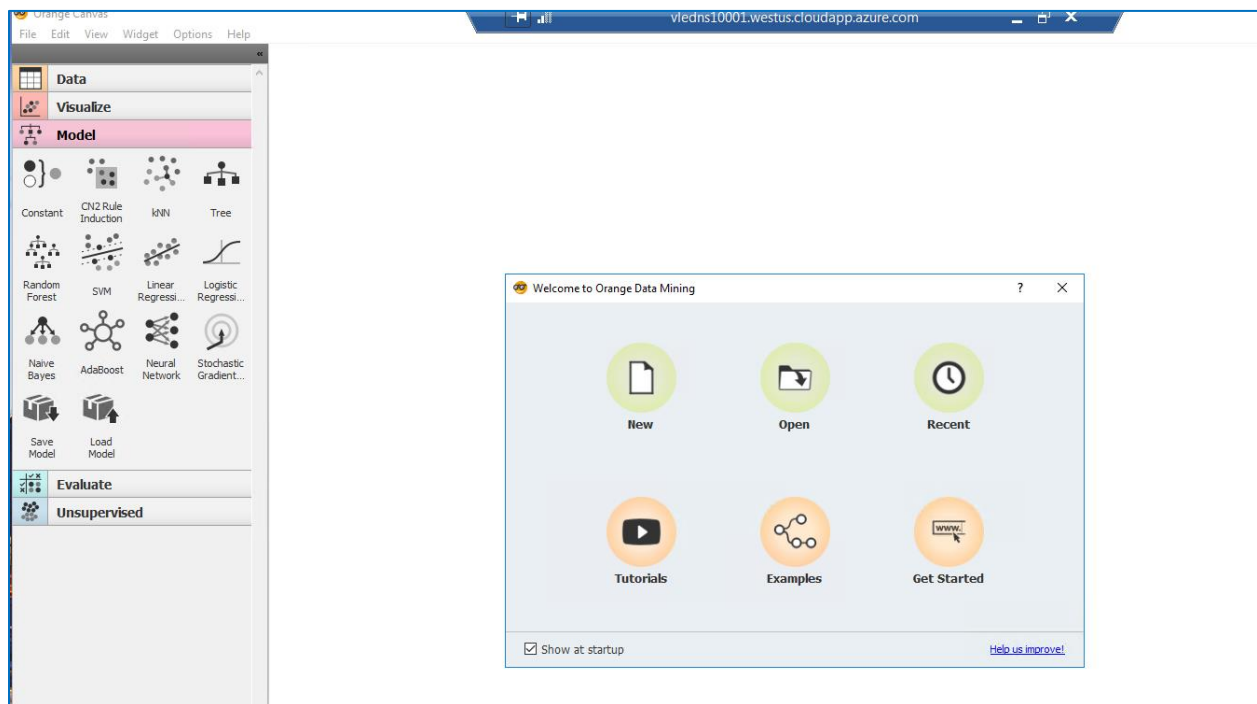
In this part of the lab, we are going to use the information that we gained from our Exploratory Data Analysis in Power BI, in order to build a predictive model. We will train and test a couple of different models and then evaluate the performance.

[Orange](#) is an open source machine learning and data visualization tool both for novice and expert. It provides interactive data analysis workflows and visualizations from within a large available toolbox, in a code free environment.

Orange is already downloaded and installed on your virtual machine, so go ahead and start it by clicking the icon either from your taskbar or the desktop shortcut



This is the first page you'll see when Orange loads:

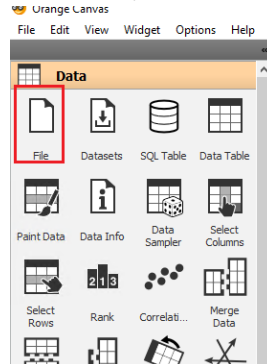


1. Select “New” and you will get a blank canvas where we are going to develop our workflow.

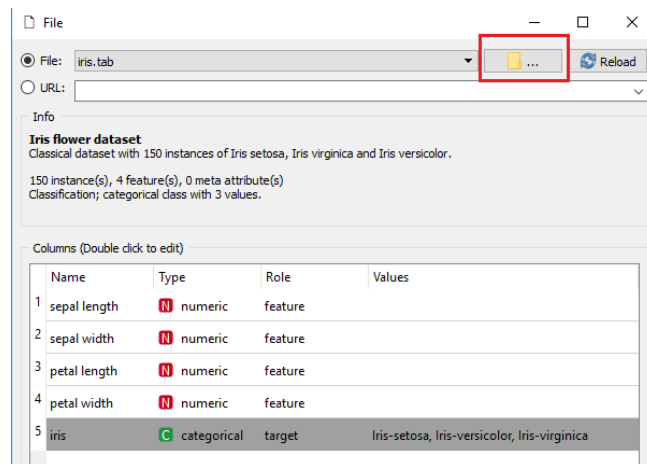
4.1.1 Loading the dataset

The first thing we need to do is load our dataset.

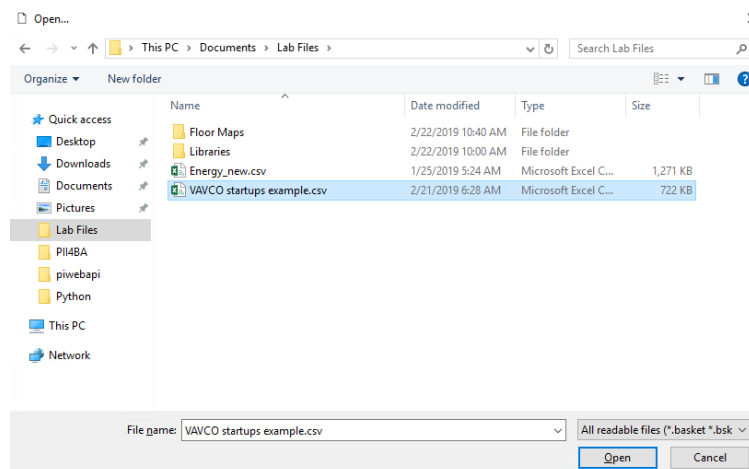
- To do that, we need to add the **File** widget from the **Data** category on the left pane.



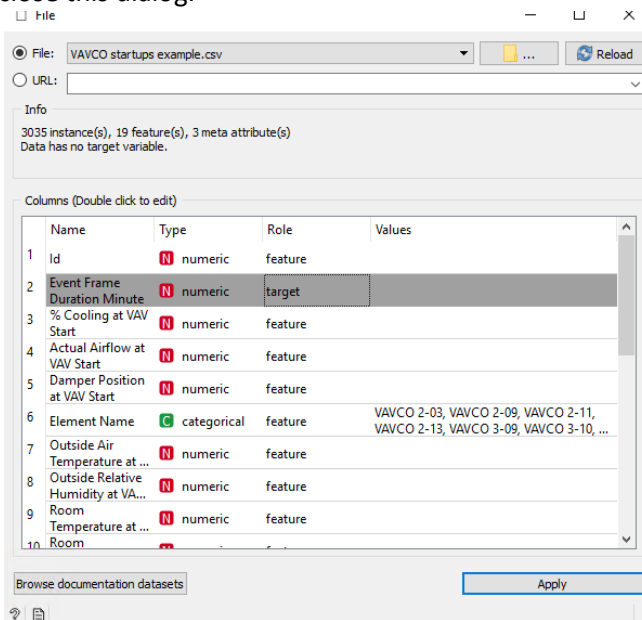
- You will then notice the File widget being added to your canvas. Double click on it to open its selection dialog and select the “Browse” symbol, next to the File tab.



- This is going to open a File Explorer, so navigate to the “**Lab Files**” folder, select the “**VAVCO startups example.csv**” file click on “**Open**”.

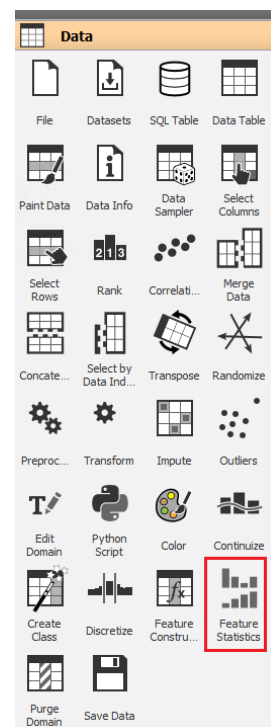
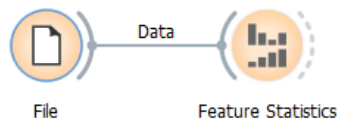


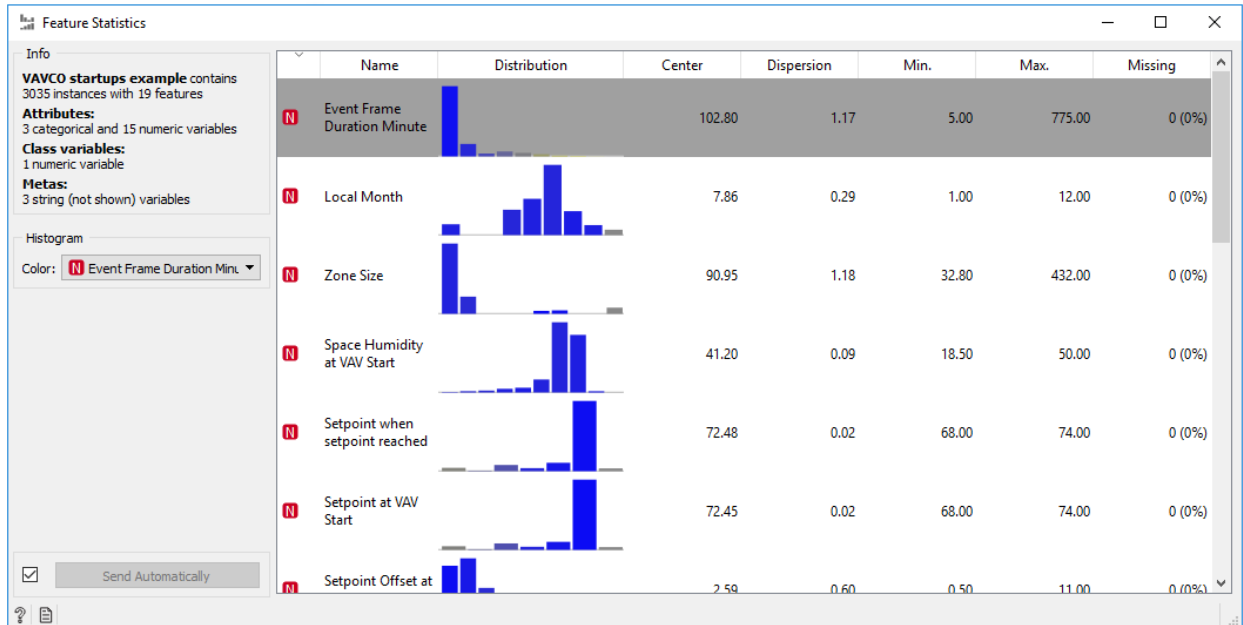
- This is going to open a table with a Preview of the columns of our dataset and some basic information about it, like the Type (numeric, categorical etc.) as well as the Role (feature, target etc.) of the different columns. Remember from our problem statement that we are trying to predict the Duration of the Cooling events in minutes, so our **Target variable** should be the **Event Frame Duration Minute**. Make sure that this is set correctly and if not, double click on the Role of the **Event Frame Duration Minute**, change it to **target** and click Apply. You can now close this dialog.



The next thing we would like to do, is to get some summary information about our variables. This functionality is provided by the **Feature Statistics** widget and we are going to add this next to our canvas and connect it to our **File** widget.

- Select the **Feature Statistics** widget again from the **Data** category.
- We now need to link those two and we can do this by dragging a link from our existing **File widget**, to the **Feature Statistics** that we just added. Have a look at the content of the **Features Statistics** now by double clicking on it.





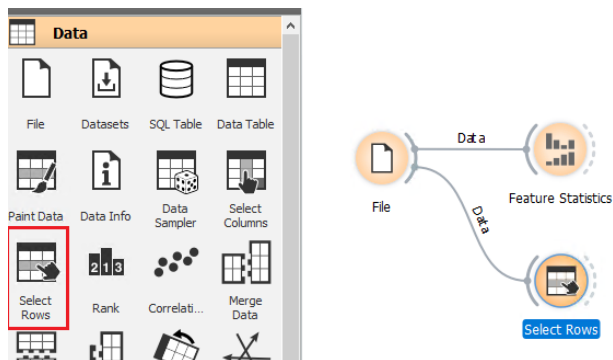
The feature statistics widget provides some basic summaries about the columns of our dataset, like the Min, Max and center of numeric columns, but it also gives you an idea about the Distribution of the values by providing a histogram. Furthermore, the last column will give you information about any missing values in your dataset. In our case it happens that we have no missing values so all values under the Missing column should be 0 (0%).

4.1.2 Filtering, Feature Engineering and Feature selection

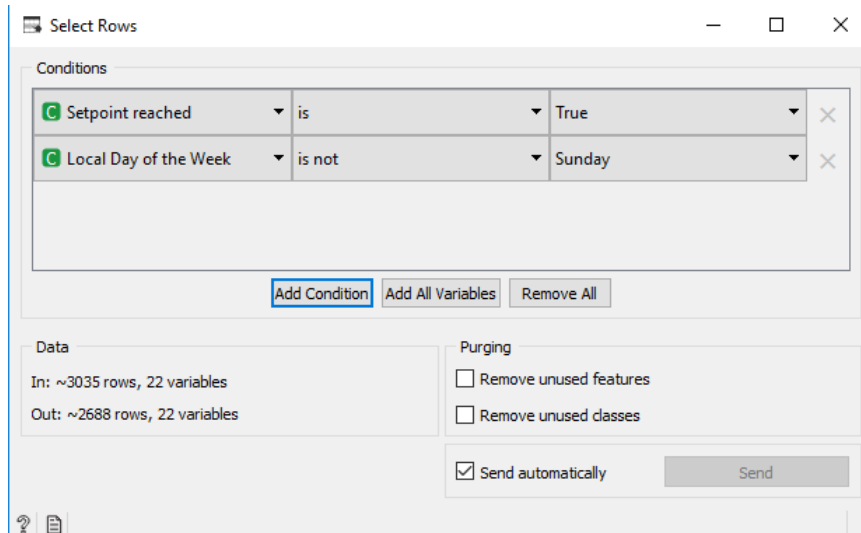
Now that we've loaded our dataset, looked at the summary statistics and verified we don't have any missing values, we are going to start preparing it for modelling. Since we are modelling on the Duration it took to reach the setpoint, we basically want to focus only on events that reached the setpoint, like we did when we were building our Power BI report.

In addition, remember that we found a few Event Frames that were created on Sunday which shouldn't be there so we need to filter those out as well.

1. Add the **Select Rows** widget from the **Data** category and connect it with the **File** widget.



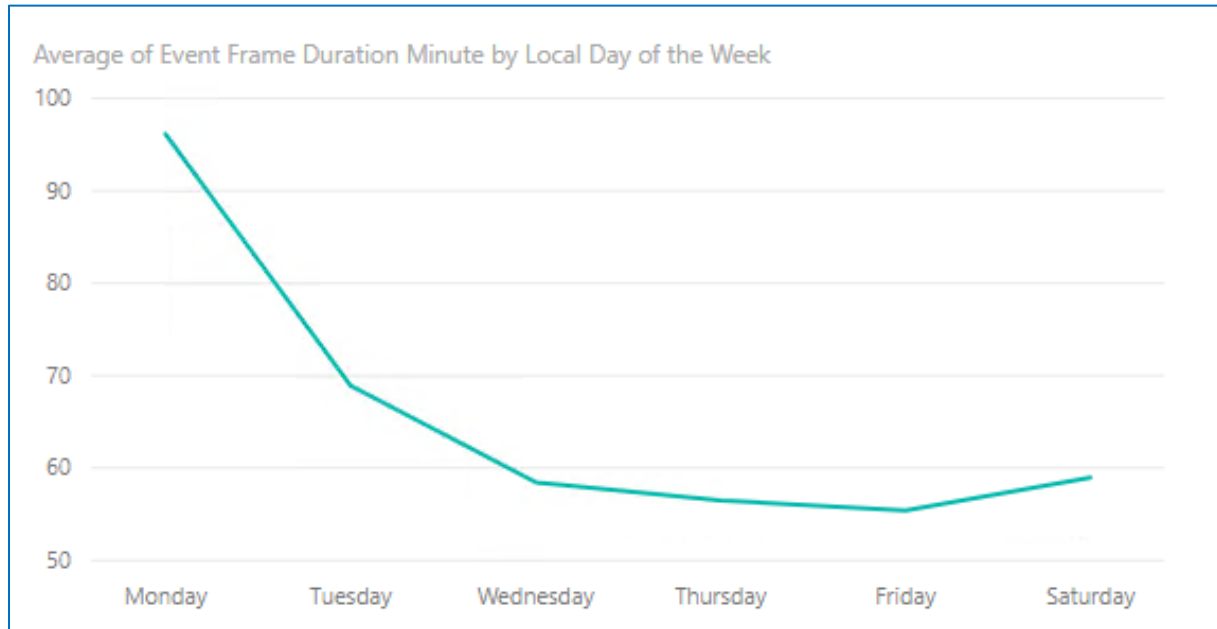
2. Double click on the **Select Rows** widget to open its configuration dialog and two conditions, based on **Setpoint reached** is **True** and **Local Day of the Week** is **not Sunday**.



In the next step we are going to perform some **Feature Engineering**.

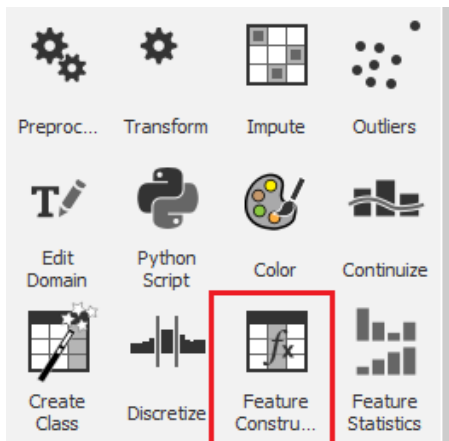
Feature Engineering is a very important step of the Data Science project lifecycle. During this step, additional relevant features are created from the existing raw features of the original dataset, in order to increase the predictive power of the learning algorithm. You can find more information about feature engineering and selection at this [Microsoft link](#).

If you remember from our Data Exploration step, we noticed that the **Event Frame Duration.Minute** variable (target variable), was very much dependent on the **Day of the week**.



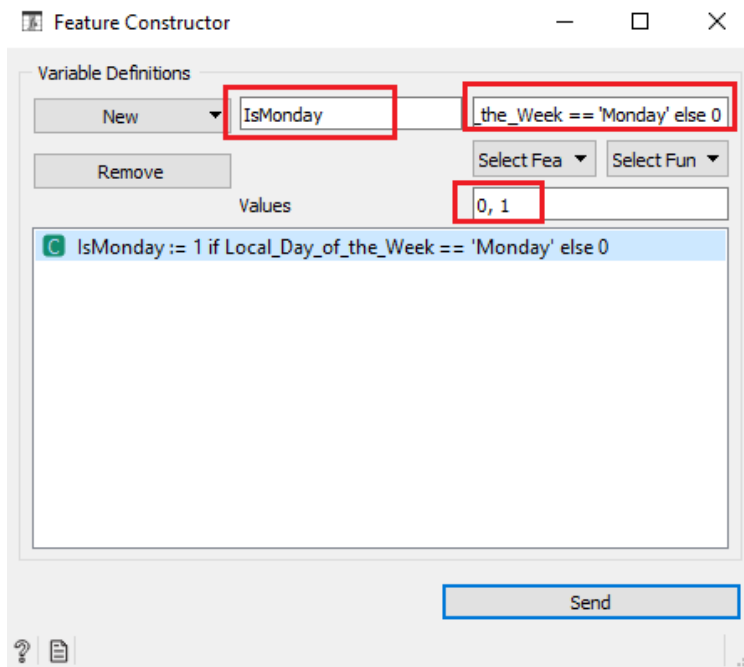
However, our observation is that the average Duration is significantly higher on Mondays, whereas it is almost constant for the other days of the week. This is a great example where we can use feature engineering to create a new variable that captures this information, since we are basically interested to check only if the day is Monday or not.

The next widget we are going to add is again from the **Data** category and is called **Feature Constructor**.

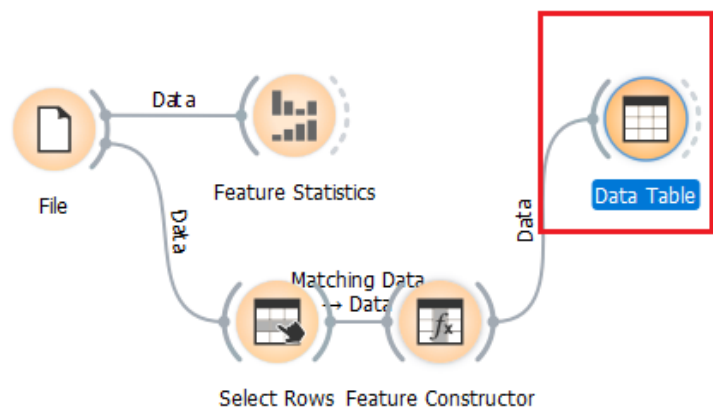
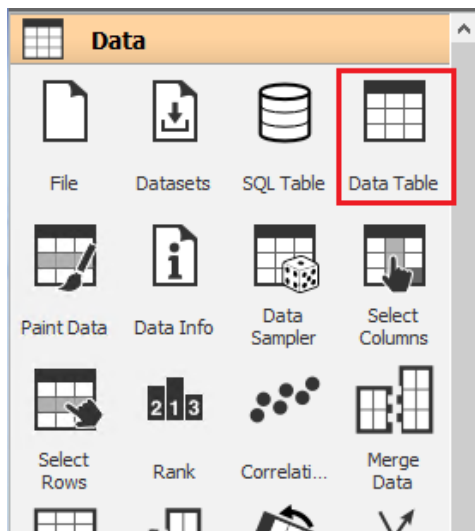


3. Connect the **Feature Constructor**, to the **Select Rows** widget, like we've done previously, and double click on it to open its configuration dialog.
4. Under the **Variable Definitions** select **New > Categorical** and type **IsMonday** in the field next to it. This is going to be our new variable name. Type the following expression in the **Expression** field: **1 if Local_Day_of_the_Week == 'Monday' else 0**

5. Finally, type 0, 1 in the **Values** field and click **Send**.



6. We now want to make sure that our variable was created correctly, so let's connect a **Data Table** widget from the **Data** category to inspect our data.



If you open the Data Table widget we just added and scroll through the first few values, you should see something similar to the following:

Data Table

Info
2688 instances (no missing values)
19 features (no missing values)
Continuous target variable (no missing values)
3 meta attributes (no missing values)

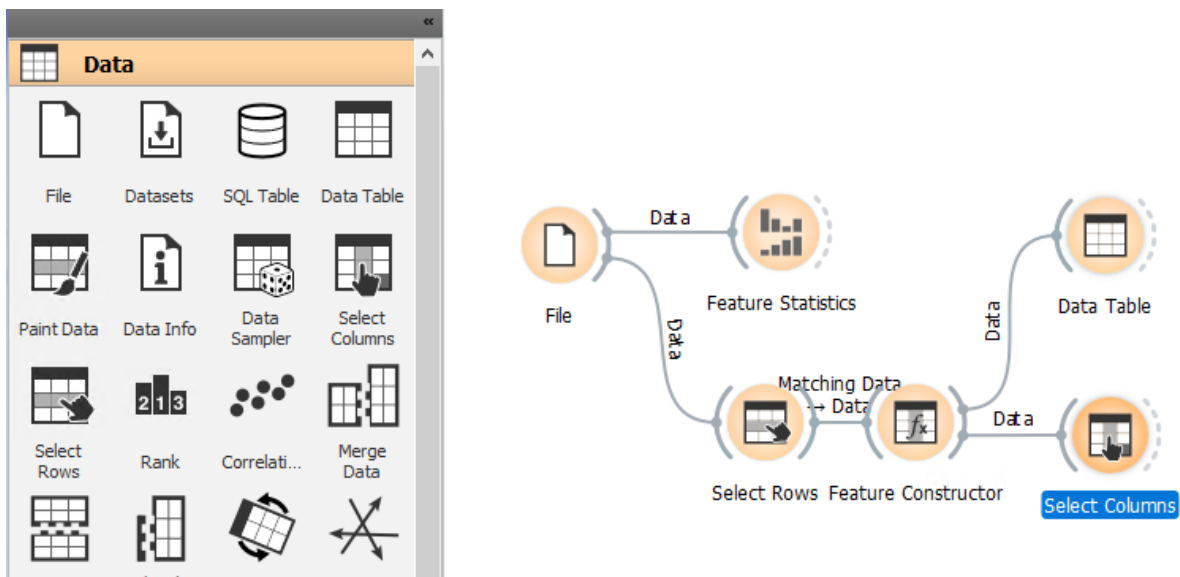
Variables
☒ Show variable labels (if present)
☐ Visualize numeric values
☒ Color by instance classes

Selection
☒ Select full rows

	int	Humidity at VAV	Zone Size	Local Month	Local Day of the Week	IsMonday
158	72	40.50000000000000	39.4	6	Friday	0
159	73	39.50000000000000	37.8	6	Friday	0
160	73	40.00000000000000	39.4	6	Friday	0
161	73	42.50000000000000	40.6	6	Friday	0
162	72	40.50000000000000	268.0	6	Friday	0
163	70	41.50000000000000	40.4	6	Friday	0
164	73	39.00000000000000	432.0	6	Friday	0
165	73	41.50000000000000	38.5	6	Friday	0
166	73	40.00000000000000	38.4	6	Monday	1
167	73	40.50000000000000	96.3	6	Monday	1
168	73	40.00000000000000	39.6	6	Monday	1
169	73	40.00000000000000	39.3	6	Monday	1
170	73	39.50000000000000	40.6	6	Monday	1
171	73	39.50000000000000	116.0	6	Monday	1
172	73	41.00000000000000	38.7	6	Monday	1
173	73	40.00000000000000	38.2	6	Monday	1
174	73	41.00000000000000	32.8	6	Monday	1
175	73	40.00000000000000	39.6	6	Monday	1
176	72	40.50000000000000	39.4	6	Monday	1

Now we have filtered our data, we also performed our feature engineering and we finally want to focus only on the features we are going to use for building our model and those will be based on the conclusions of our data exploration. Our Data Exploration showed that the features that seem to be highly correlated with our target variable (**Event Frame Duration Minute**) were the **Setpoint Offset at VAV start** and **% Cooling at VAV start**. We would like to use just those 2 features, plus the one that we created in the previous step, the **IsMonday**.

- Let's add the **Select Columns** widget now from the **Data** category and connect it to the **Feature Constructor** widget.



8. Under the **Feature** field we only want to have **Setpoint Offset at VAV start, % Cooling at VAV start and IsMonday**. Make sure that under the **Target variable** field you have **Event Frame Duration Minute** and let's add **VAVCO startup** and **Element Name** under the **Meta Attributes** field (we are going to need those later to evaluate our results).

Select Columns

Available Variables

Filter

- N Id
- N Actual Airflow at VAV Start
- N Outside Air Temperature at VAV Start
- N Outside Relative Humidity at VAV Start
- N Room Temperature at VAV Start
- N Room Temperature when setpoint reached
- N Setpoint Offset at end time
- N Setpoint at VAV Start
- C Setpoint reached
- N Setpoint when setpoint reached
- N Space Humidity at VAV Start
- N Local Month
- S Event Frame Start Time (Local) TimeStamp
- N Damper Position at VAV Start
- C Local Day of the Week
- N Zone Size
- S Event Frame End Time (Local) TimeStamp

Features

Filter

- N Setpoint Offset at start time
- C IsMonday
- N % Cooling at VAV Start

Target Variable

- N Event Frame Duration Minute

Meta Attributes

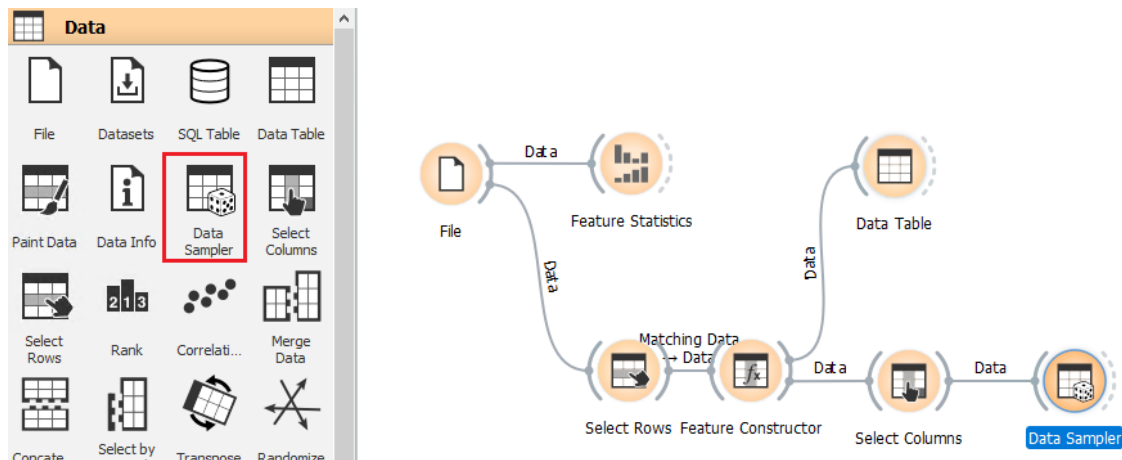
- S VAVCO startup
- C Element Name

Reset ☒ Send Automatically

4.1.3 Splitting the data into Training and Test sets

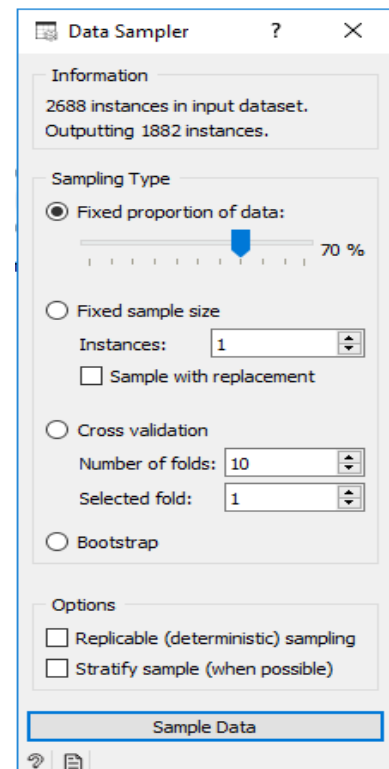
Before training our model, we need to split our data into training and test sets. This is a common task in the model building process. We need to train our model on a training dataset and then test the performance of the model with some data that it hasn't "seen" before. Since we only have one dataset, it is common to randomly split it in 2, using one subset to train the model and the other subset to test the model. There are various methods for splitting and sampling a dataset but in our case we will use a 70% random sample for our training set and the rest for our test set.

1. Add the **Data Sampler** widget from **Data** category and connect it to the **Select Columns** widget.



2. Open the configuration dialog of the configuration dialog of the **Data Sampler** and make sure that we are using a **Fixed proportion of data** set to **70%**. Then, select **Sample Data**.

We have now split our dataset into a Training set that will be used to train the model and Test set that will be used to evaluate the model.



4.1.3 How to choose the right model

Every machine learning algorithm has its own style, advantages and disadvantages. For a specific problem, more than one algorithm may be appropriate and one algorithm may be a better fit than others. However, it's not always possible to know beforehand which the best fit is. An appropriate strategy is usually to try one algorithm, and if the results are not yet satisfactory, try the others. There are however some basic rules on which type of models to try with specific types of problems.

You can find plenty of online documentation regarding available machine learning models and when to choose one over the others. Personally, I use just two main question when it comes to deciding which way to go about picking a model.

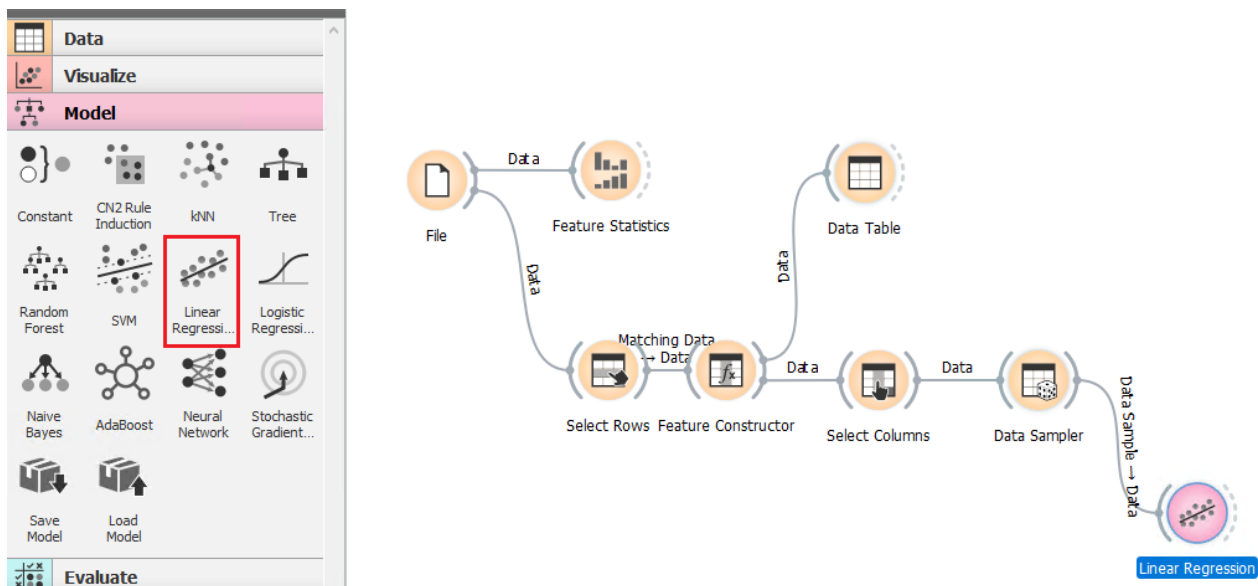
Trying to predict specific values? → Regression models

Trying to predict a category? → Classification models

There are of course other types of questions in machine learning but those are the two main categories that you would normally see more often and they belong to the main category of **Supervised Machine Learning**.

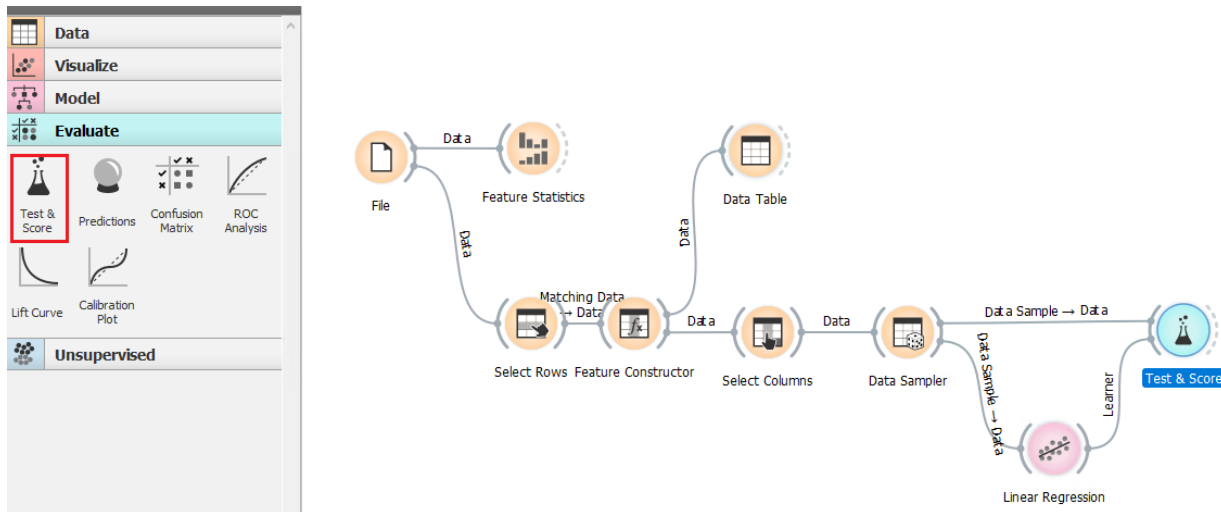
Since in our case we are trying to predict the time it takes to reach the temperature setpoint, we are actually trying to predict specific values which by looking at the two main questions I listed above, leads us to start by using a regression model. The first and simplest regression model that we are going to use, is **Linear Regression** and we are going to use this since our Exploratory Data Analysis phase also showed that there is some linear relationship between the **target** and the **predictor** variables.

1. Let's add the **Linear Regression** widget which this time is located under the **Model** category and connect it to the **Data Sampler** widget.



If you open the configuration dialog of the **Linear Regression** widget, you'll see the different available options, but in this case we are just going to use the Default **No Regularization**.

- The next thing we need to add is the **Test & Score** widget which is located under the **Evaluate** category and connect it both to the **Linear Regression** and **Data Sampler** widgets.



- If you double click on the **Test & Score** widget you will see related information about model metrics (Mean Squared Error, Root Mean Squared Error, Mean Average Error and R squared), based on the training dataset.

The 'Test & Score' window displays the following settings and results:

Sampling

- ☐ Cross validation
 - Number of folds: 10
 - ☒ Stratified
- ☐ Cross validation by feature
 - Element Name
- ☐ Random sampling
 - Repeat train/test: 10
 - Training set size: 66 %
 - ☒ Stratified
- ☐ Leave one out
- ☒ Test on train data
- ☐ Test on test data

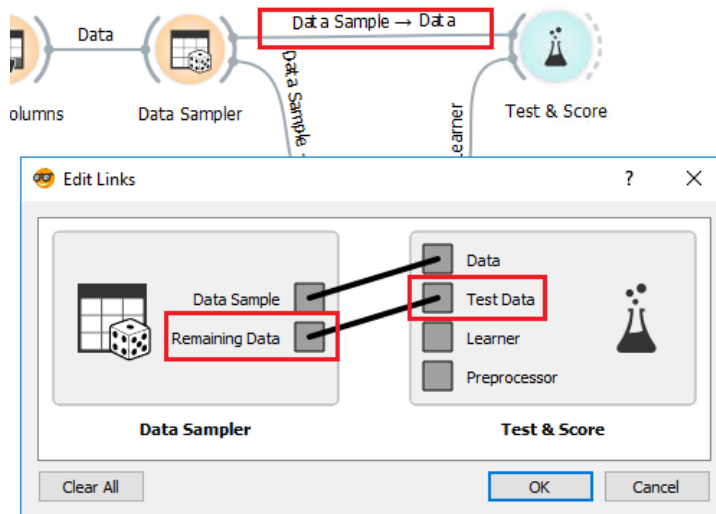
Evaluation

Click on the table header to select shown columns

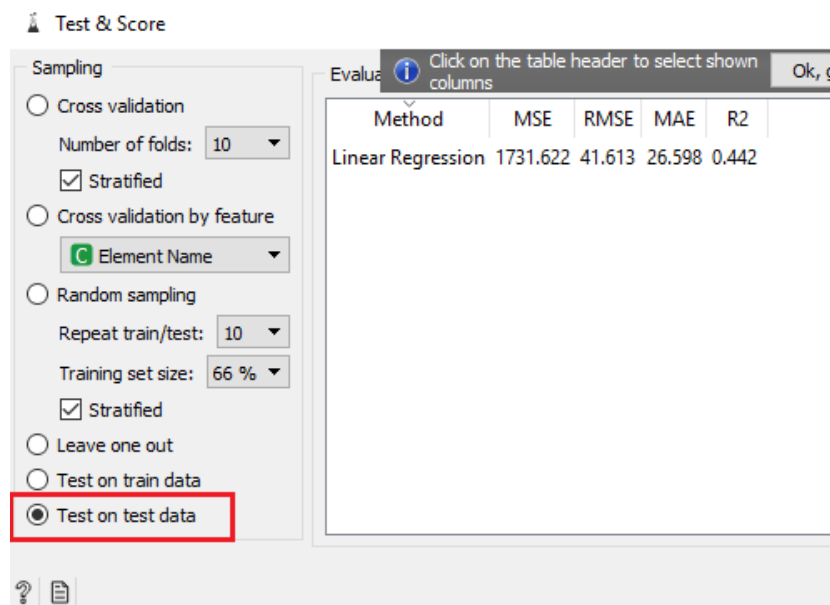
Method	MSE	RMSE	MAE	R2
Linear Regression	1402.960	37.456	24.054	0.509

In order to evaluate our model, we also need to test on the Test set as well though (remember we split our data into training and test sets).

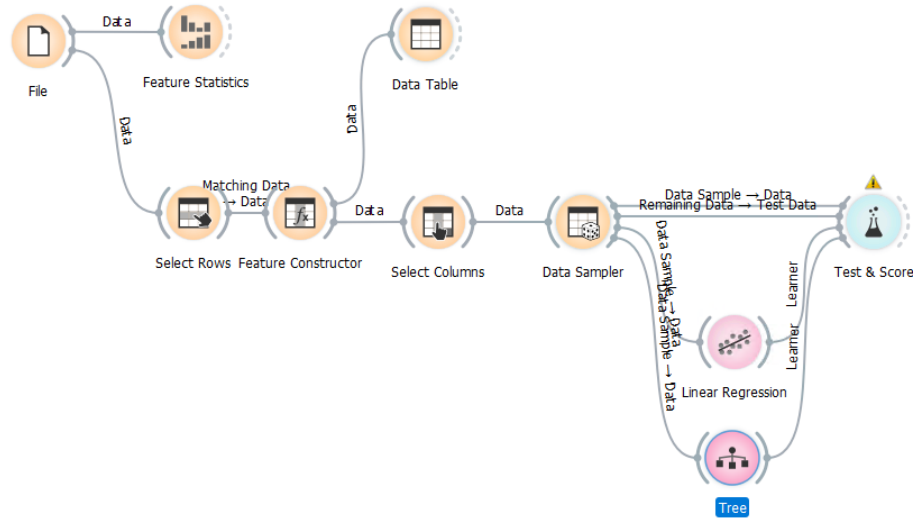
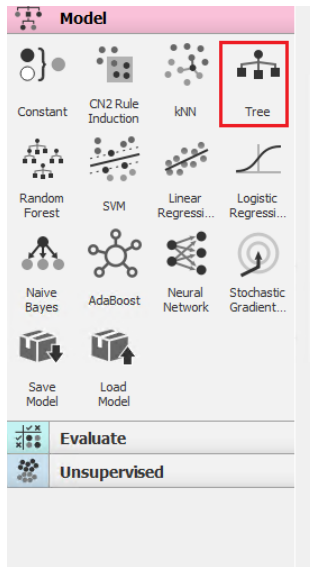
4. In order to do that, just double click on the connector line between the **Data Sampler** and the **Test & Score** widgets. This will open a dialog where you can connect the **Remaining Data** from the **Data Sampler**, with the **Test Data** from the **Test & Score**.



5. Now open the **Test & Score** configuration panel again and switch from **Test on Train data** to **Test on test data** to see how our model performs on the test dataset.



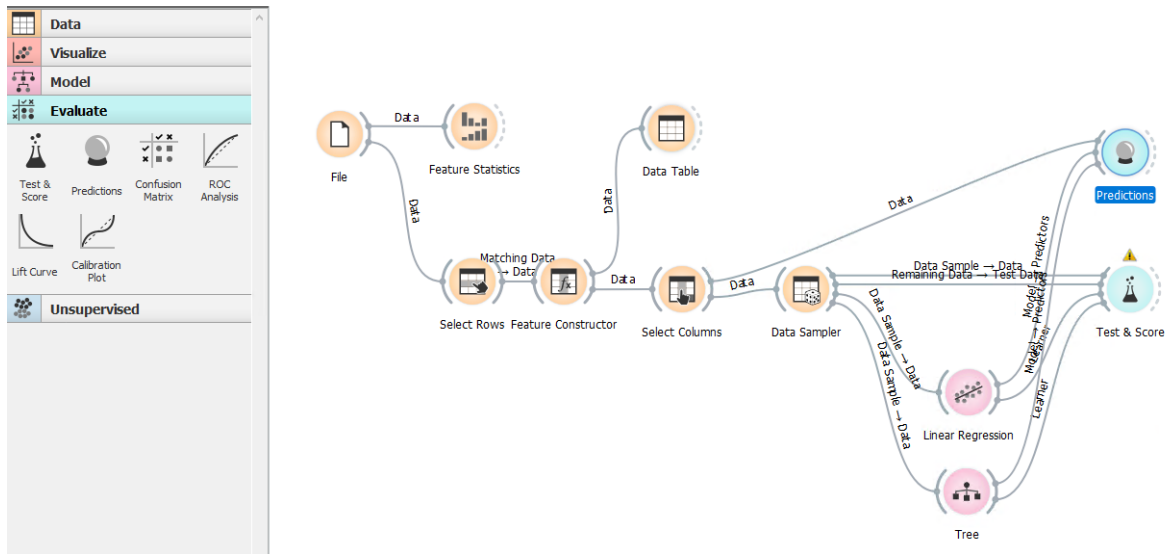
- As a next step, we can also add other types of models into our analysis and compare their performance. In this case we are going to add a **Decision Tree**, but feel free to add other regression models as well and see how well they perform.



4.1.4 Results Evaluation

At this point we've built and tested a couple of models but the next step would be to actually compare how the model would perform, compared to the current system that is handling the VAVCO units startup. To do that, we are going to first use the **Predictions** widget from the **Evaluate** category.

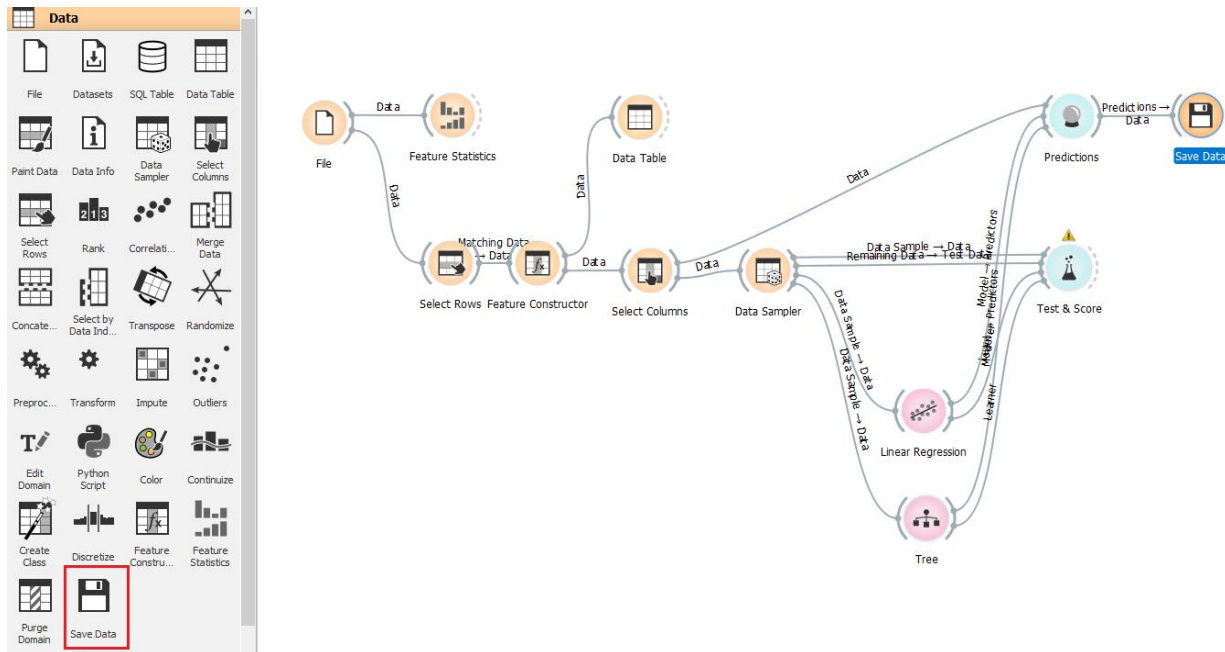
We need to connect the **Predictions** widget to the **Linear Regression** and **Tree** widgets in order to use those models for predictions. We also need to connect the **Predictions** widget to the **Select Columns** widget in order to use our whole dataset for predictions (we will generate predictions on our historical dataset, splitting into training and test was just in order to train our models).



The contents of the **Predictions** widget is a table showing the predictions from the two models that we used (Linear Regression and Decision Tree), as well as the actual values of the event Frame Duration and the rest of the features.

Our final step will be to save those into a file and load it to Microsoft power BI in order to visually compare the results from our model, with the data from the current system.

1. Add the **Save Data** widget from the **Data** category and connect it to the **Predictions** widget.



2. Open the configuration dialog of the **Save Data** widget, choose **Comma-separated values** as the File Type and **Save As...** Navigate to your **Lab Files** folder and give your file a name like **Model Results.csv**.

We now want to load the **Model Results.csv** file into Power BI to compare the performance of the model, with the real data.

Power BI Results comparison

- Go back to your **Power BI report** and create a new page (Page 3). From the menu ribbon, click on **Get Data** to expose the dropdown (right). Select **Text/CSV**. This will open the **Get Data** dialog.
- Navigate to the **Lab Files** folder and select the **Model Results.csv** file that we created in the previous step (Alternatively, you can also load “*Model Results Sample.csv*” if you didn’t get to create the output file from the previous exercise). Notice that the preview shows that the column names are not correct and there are also some extra rows in our dataset. Load the file as it is for now and we’ll fix those later.

Model Results example.csv

File Origin: 1252: Western European (Windows) | Delimiter: Comma | Data Type Detection: Based on first 200 rows

Column1	Column2	Column3	Column4	Column5	Column6
Setpoint Offset at start time	IsMonday	% Cooling at VAV Start	Event Frame Duration Minute	VAVCO startup	Element
continuous	discrete	continuous	continuous	string	discr
			class	meta	meta
2.5	0	40.3749694824219	55.0	VAVCO startup - VAVCO 3-09 - 2018-06-06 07:01:23.880	VAVCO
2.0	0	33.4999809265137	40.0	VAVCO startup - VAVCO 5-12 - 2018-06-06 07:01:23.911	VAVCO
1.5	0	26.6416606903076	35.0	VAVCO startup - VAVCO 5-13 - 2018-06-06 07:01:23.926	VAVCO
2.0	0	33.066650390625	60.0	VAVCO startup - VAVCO 5-10 - 2018-06-06 07:01:23.973	VAVCO
2.0	0	33.7166404724121	25.0	VAVCO startup - VAVCO 6-06 - 2018-06-06 07:01:24.036	VAVCO
2.0	0	33.4666404724121	35.0	VAVCO startup - VAVCO 5-11 - 2018-06-06 07:01:24.208	VAVCO
1.0	0	19.8999996185303	25.0	VAVCO startup - VAVCO 4-17 - 2018-06-06 07:01:24.801	VAVCO
1.5	0	26.7499904632568	100.0	VAVCO startup - VAVCO 4-16 - 2018-06-06 07:01:24.911	VAVCO
1.5	0	26.7249908447266	30.0	VAVCO startup - VAVCO 5-09 - 2018-06-06 07:01:24.942	VAVCO
1.5	0	26.7083206176758	40.0	VAVCO startup - VAVCO 2-03 - 2018-06-06 07:01:25.208	VAVCO
2.5	0	40.2999610900879	50.0	VAVCO startup - VAVCO 3-15 - 2018-06-06 07:01:25.348	VAVCO
2.0	0	33.4249801635742	35.0	VAVCO startup - VAVCO 3-10 - 2018-06-06 07:01:25.817	VAVCO
2.0	0	33.5166397094727	95.0	VAVCO startup - VAVCO 3-16 - 2018-06-06 07:01:25.817	VAVCO
2.0	0	33.6333084106445	30.0	VAVCO startup - VAVCO 4-15 - 2018-06-06 07:01:25.817	VAVCO
2.5	0	40.4999618530273	50.0	VAVCO startup - VAVCO 6-09 - 2018-06-06 07:01:25.833	VAVCO
2.5	0	40.4749603271484	65.0	VAVCO startup - VAVCO 6-10 - 2018-06-06 07:01:25.833	VAVCO
2.5	0	40.4749603271484	45.0	VAVCO startup - VAVCO 6-11 - 2018-06-06 07:01:25.833	VAVCO

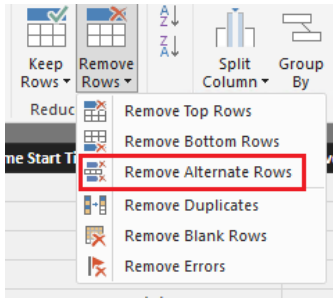
Load Edit Cancel

- To fix those issues, select the **Edit Queries** icon from the ribbon menu. This will open the query editor.



icon from the ribbon menu. This will open

- Make sure that on the left pane you have selected the **Model Results** data table (we need to make changes to the new table we just imported) and select the **Remove Rows** icon from the ribbon menu (under the **Reduce Rows** category) and then **Remove Alternate Rows**



- In the dialog that opens, add the following options (we want to remove the second and third row and keep all the rest).

Remove Alternate Rows

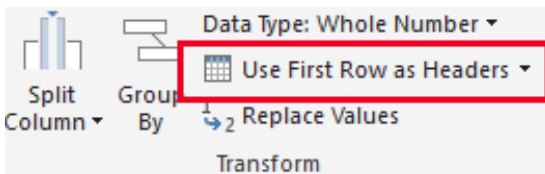
Specify the pattern of rows to remove and keep.

First row to remove
2

Number of rows to remove
2

Number of rows to keep
3000

- Finally, select **Use First Row as Headers** from the **Transform** category on the ribbon menu.



- You can now close the Query Editor and select **Yes** when you are asked to **Apply your changes**.
- We now have two tables into our report, the **VAVCO startup example** and the **Model Results**. If you check the **Relationships** from the menu on the left, you will notice that those tables are connected based on their common column, called **VAVCO startup**.
- First, let's add the **Setpoint reached** from the **VAVCO startup example** table as a page level filter (like we did in the previous report) and filter on **True**, since we have only modeled the events when the setpoint was reached.

In this report, we would like to compare the **Total Lost Hours** that we calculated previously, with the losses from the prediction errors of our model. To do that, we first need to calculate the Model Losses.

12. Right-click on the table name **Model Results**, select **New column** and add the following text in the formula bar:

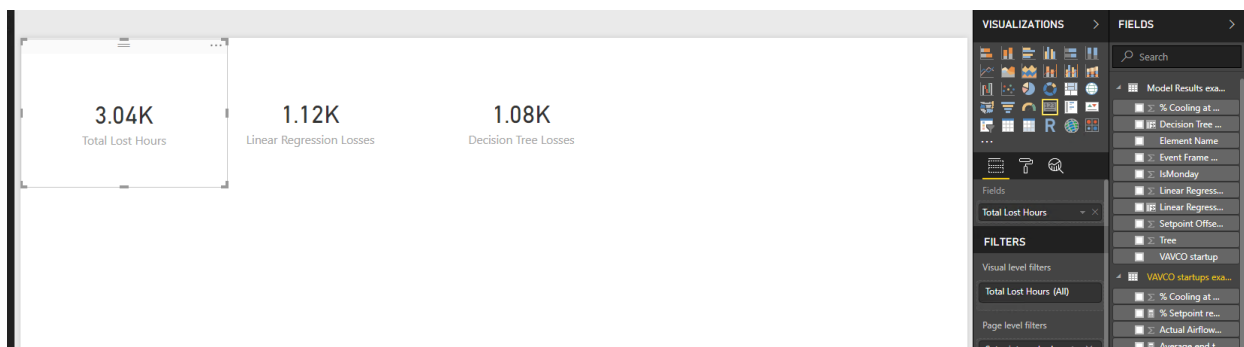
Linear Regression Losses = abs('Model Results example'[Event Frame Duration Minute]-'Model Results example'[Linear Regression])/60

13. We also want to calculate the losses from our Decision Tree model, so let's add one more column with the following calculation:

Decision Tree Losses = abs('Model Results example'[Event Frame Duration Minute]-'Model Results example'[Tree])/60

14. Let's add those calculations now into our report to compare the Losses. Click a blank space on the report canvas. Select a **card** visual from the pallet in the visualization pane. Add the **Total Lost Hours** from the fields list of **VAVCO startup example** into the card.

15. Add two more card visuals for the **Linear Regression Losses** and the **Decision Tree Losses** from the **Model Results Table**.

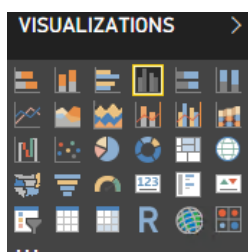


It looks like both our models would reduce the losses significantly. The decision Tree looks better than the Linear Regression in that case, but your results could be slightly different.

Let's see how the models perform on a VAVCO o VAVCO basis as well.

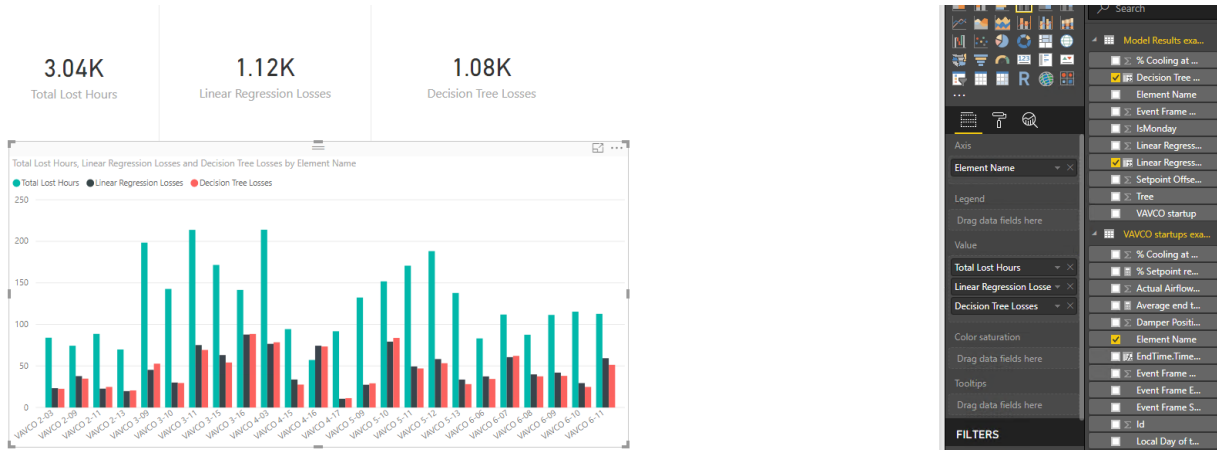
Clustered Column Chart

16. Click a blank space on the report canvas. Select a **Clustered column chart** visual from the pallet in the visualization pane.



17. Add the **Element Name** to the **Axis** field of the clustered column chart and the **Total Lost Hours** field from the fields list of **VAVCO startup example**, in the **Values** field of the column chart. Also add the **Linear Regression Losses** and the **Decision Tree Losses** from **the Model Results** table.

Your stacked column chart should look similar to the one below:



This proves that both our models would significantly outperform the system that is currently handling the Building startup. In terms of **Total Lost Hours** we can see that our models would reduce the losses to about 1/3 of the original ones. Moreover, in terms of evaluating on a unit by unit basis, using one of the models that we developed would again significantly reduce the losses in almost every case.

This shows how sometimes even an average model ($R^2=0.45$) can have great results when used in production. Sometimes we might get caught on improving model statistics and try to optimize a model as much as possible, or even reject models because of just looking at model metrics, instead of actually comparing the model performance against the process we are trying to improve.

Part 5 – Deployment



Discussion

We have come to the point where we trained and evaluated our model. The next step in our Data Science project lifecycle (following the CRISP-DM methodology) is the part of **Deployment**. This is where we want to have our model run in real time and be able to control the startup of the VAVCO units based on the model predictions.

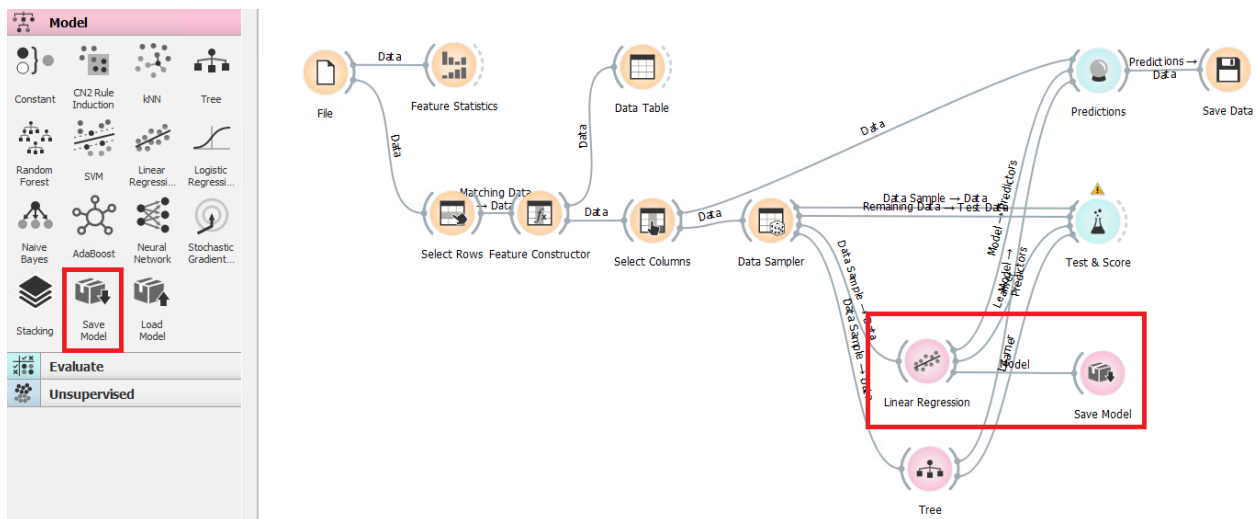
There are a lot of different options for model deployment of course depending on different factors like complexity of the model, model performance etc. Simpler models like linear regression for example could even be deployed in PI, for some other more complex models you might need to turn to a third party advanced analytics platform.

In all cases however, **PI provides the ability to store the predictions** (potentially as **future data**), closing the loop of a Data Science project lifecycle and that's what we are going to show in this section.

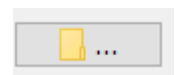
Save Model from Orange Workflow

In this example we are going to operationalize the Linear Regression that we trained and evaluated in the previous step.

1. Go back to your Orange canvas where we have developed our workflow and add the **Save Model** widget from the **Model** category and connect it to the **Linear Regression** widget.



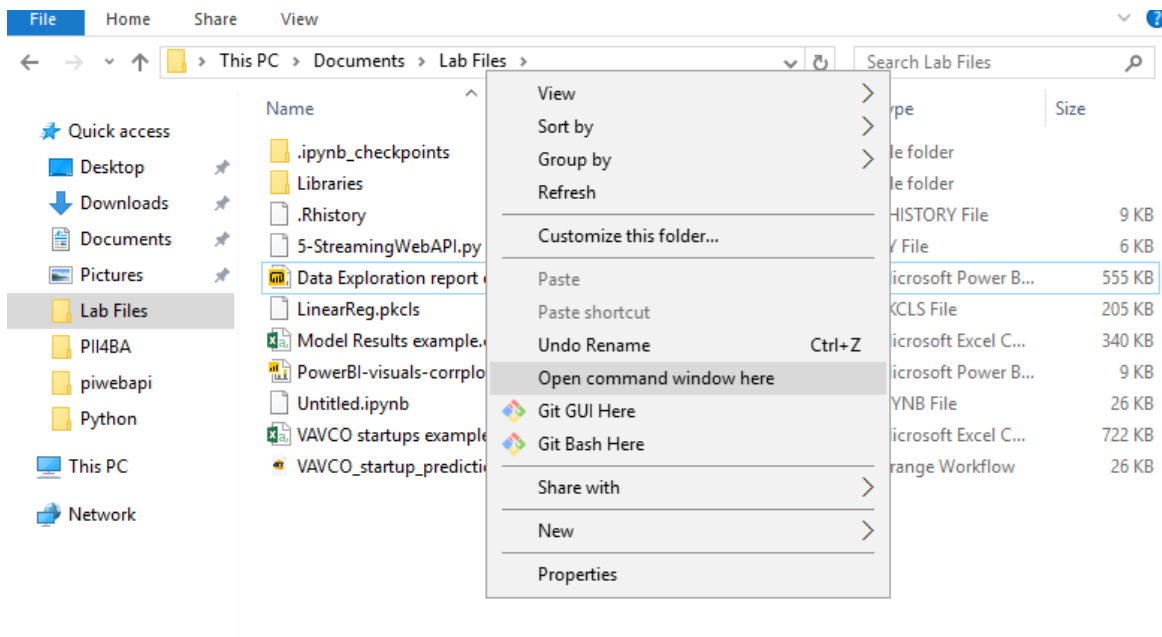
2. Now open the configuration dialog of the **Save Model** widget, select the folder symbol, navigate to your Lab Files folder, type a name for you model (e.g. **MyModel**) and click Save. This will create a file that holds your Linear Regression model configuration and can be used for Real Time predictions.



Use python script to store predictions in PI

For the purposes of this example, we have created a python script that uses PI Web API to read real time data from PI, feed this to our Linear Regression Model (that we saved in the previous step), produce the predictions and finally store those back in PI under the **Predicted Cooling Time** attribute.

3. The script is called **5-StreamingWebAPI.py** and it is located in your **Lab Files** folder. To run the script, Navigate to you **Lab Files** folder, click somewhere in the white space and then **shift + right mouse click** and select **Open command window here**



4. On the command window that opens type the following:

`python 5-StreamingWebAPI.py <password>` (where <password> is the password you used to log into your VM) and hit Enter.

After a few seconds you should see something similar to this in the command window:

This script runs a pre-exported model called “LinearReg.pkcls” that you can see in your **Lab Files** folder.

5. If you want to use your own model instead (the one you saved in the previous step), open the script and modify the parameter `model_path = 'your_model_name'`.

```
Time to reach setpoint
Time: Wed Mar 6 04:16:55 2019
VAVCO 2-03: 0.0 minutes
VAVCO 2-09: 58.5 minutes
VAVCO 2-11: 0.0 minutes
VAVCO 2-13: 22.7 minutes
VAVCO 3-09: 58.5 minutes
VAVCO 3-10: 22.7 minutes
VAVCO 3-11: 4.8 minutes
VAVCO 3-15: 22.7 minutes
VAVCO 3-16: 4.8 minutes
VAVCO 4-03: 40.6 minutes
VAVCO 4-15: 22.7 minutes
VAVCO 4-16: 4.8 minutes
VAVCO 4-17: 0.0 minutes
VAVCO 5-09: 0.0 minutes
VAVCO 5-10: 0.0 minutes
VAVCO 5-11: 0.0 minutes
VAVCO 5-12: 130.1 minutes
VAVCO 5-13: 40.6 minutes
VAVCO 6-06: 0.0 minutes
VAVCO 6-07: 4.8 minutes
VAVCO 6-08: 4.8 minutes
VAVCO 6-09: 22.7 minutes
VAVCO 6-10: 22.7 minutes
VAVCO 6-11: 22.7 minutes
```

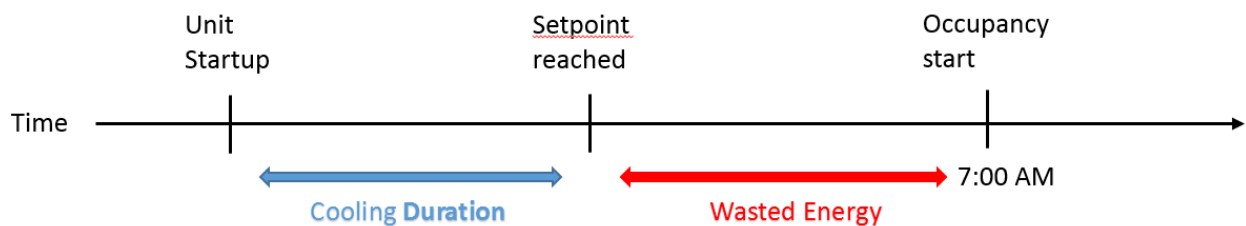

- Now navigate to your AF hierarchy in PI System Explorer and browse through your VAVCO elements. You should be able to see attribute called **Predicted Cooling Time** updated with current predictions.

VAVCO 6-06	Cooling Damper Time (seconds)	90
VAVCO 6-07	Cooling SP Offset	1 deg F
VAVCO 6-08	Occupied Setpoint Hi	74 deg F
VAVCO 6-09	Unoccupied Cooling Setpoint	85 deg F
VAVCO 6-10		
VAVCO 6-11		
Power		
Weather		
Element Searches		
Category: Forecasts		
	Predicted Cooling Time	22.703 min
Category: Heating		
	% heating	0 %
	Warmup	0



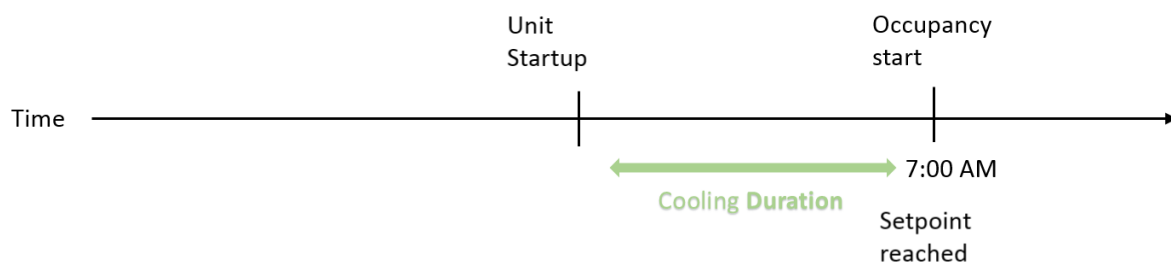
Discussion

Remember what was our initial Goal of this project? “Eliminate the Wasted Energy by having the units reach the setpoint as close as possible to 7 AM.



Discussion

Now that we have a model to predict how long it will take for a unit to reach the setpoint, we can control the startup time of those units based on the predictions. There could be either an operator or an automated process that looks into the **Predicted Cooling Time** and the Current Time and start the unit when “*Current Time + Predicted Cooling Time*” is close enough to 7 AM.



Appendix - Publish Data with PI Web API

(Optional – Demonstration)

Congratulations if you've made it this far!

This means you've covered all the material which was in the scope of the lab.


The following part goes through a bonus exercise, showing an example of accessing PI data directly from a Jupyter notebook and performing some Data Exploration with it.

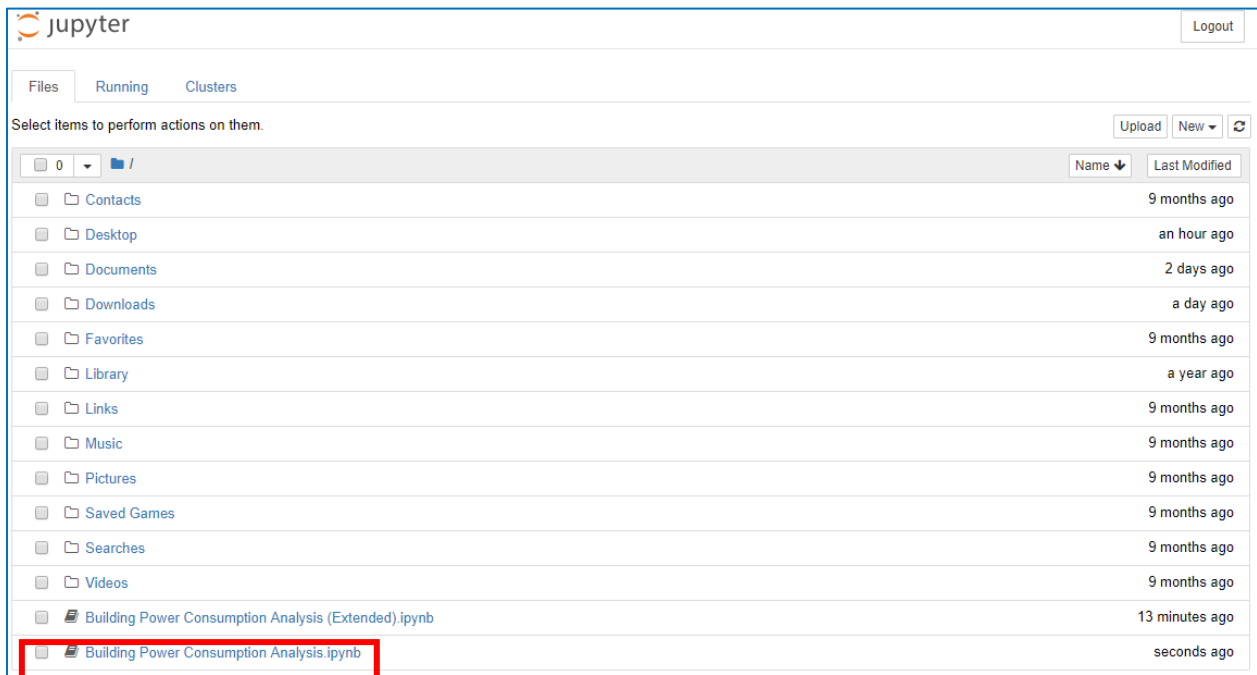
(If you're not familiar with Jupyter Notebooks, it is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text.)

In order to do that, we have used PI Web API along with some helper functions that we created for the scope of this exercise. If you're interested, you can find the sample code that we've used under the following folder path in your VM:

C:\Users\student01.PISCHOOL\Documents\Lab Files\Libraries\Python

Import Power Consumption Data in Jupyter Notebooks

1. Open Jupyter Notebooks by selecting the  shortcut from your desktop or the icon in your taskbar. The screen below shows the Jupyter Notebooks home page. We are interested in the file named "Building Power Consumption Analysis.ipynb". By clicking this file, you will get to a page where we have already typed some code.



The screenshot shows the Jupyter Notebook web interface. At the top, there's a 'jupyter' logo and a 'Logout' button. Below that, there are tabs for 'Files', 'Running', and 'Clusters'. A message says 'Select items to perform actions on them.' with 'Upload', 'New', and a refresh icon. A file browser shows a list of files and folders. The file 'Building Power Consumption Analysis.ipynb' is highlighted with a red box.

Name	Last Modified
Contacts	9 months ago
Desktop	an hour ago
Documents	2 days ago
Downloads	a day ago
Favorites	9 months ago
Library	a year ago
Links	9 months ago
Music	9 months ago
Pictures	9 months ago
Saved Games	9 months ago
Searches	9 months ago
Videos	9 months ago
Building Power Consumption Analysis (Extended).ipynb	13 minutes ago
Building Power Consumption Analysis.ipynb	seconds ago

The objective of this exercise is to demonstrate the ability to access AF data directly from Python, by running through a simple example.

2. Place your cursor on the top cell and run the code by clicking on the **Run** button on the top of the page. Alternatively, you can do the same with the shortcut “Shift + Enter”
3. Make sure to replace the **<password_placeholder>** with the password that you used to log in to your VM

The screenshot shows a Jupyter Notebook titled "Building Power Consumption Analysis" with a "Last Checkpoint: a few seconds ago (autosaved)" status. The interface includes a top bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help" menus, along with a "Trusted" status and "Python 3" version indicator. The notebook contains three code cells:

Load required libraries

```
In [ ]: import sys
sys.path.insert(0, r'C:\Users\student01.PISCHOOL\Documents\Lab Files\Libraries\Python\osisoft\pidevclub\piwebapi')

from WebAPIHelper import *
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
```

Create the PI Web API connection

```
In [ ]: webapiurl = "https://localhost/piwebapi"
dataarchive = 'PISRV01'
afserver = 'PISRV01'
afdatabase = 'Lab_Building_Data'
password = <password_placeholder> # Replace <password_placeholder> with the password you used to log into your VM

client = PIClient(webapiurl, dataarchive, afserver, afdatabase, password)
```

Load Power and Weather data

```
In [ ]: # We are creating a function to retrieve interpolated values for multiple AF attributes

def interpolated_values(paths, starttime, endtime, interval):
    for path in paths:
        if paths.index(path)==0:
            df = client.data.get_interpolated_values(path, start_time=starttime, end_time=endtime, interval=interval,
                                                    selected_fields="items.value;items.timestamp")
```

4. After one cell has run you can move to the next one by clicking “Run” again and observe the results. There are comments above each line of code that explain what each part of the code does.

In short, we are creating a connection to PI Web API and directly retrieving recorded values from the PI System. The purpose of this sort example is to perform some Data Exploration on the building’s Power Consumption.

5. Go through the provided code cells for some initial data exploration and feel free to add your own analysis, or even try to develop a model that predicts Power Consumption!



Have an idea how to
improve our products?
**OSIsoft wants to hear
from you!**

<https://feedback.osisoft.com/>





Save the Date!

OSIsoft PI World Users Conference in Gothenburg, Sweden. September 16-19, 2019.

Register your interest now to receive updates and notification early bird registration opening.

https://pages.osisoft.com/UC-EMEA-Q3-19-PIWorldGBG-RegisterYourInterest_RegisterYourInterest-LP.html?_ga=2.20661553.86037572.1539782043-591736536.1533567354

