**Problem:**

Synergy is Western Australia’s largest energy supplier to the South West Interconnected System (SWIS). The energy market operates that all individual facilities are provided with ‘availability payments’. These are the payments that are provided to generators as an incentive to stay available. The only circumstance in which availability payments are with-held or taken away is in the case where the generator experiences an unplanned outage which is commonly known as a **forced** outage. In the case a forced outage occurs, Reserve Capacity Payments are issued which are financial penalties generators must pay for not being available. Depending on the time of day and the unit capacity, penalties can range from $0 AUD up to $1,000,000 + depending on the severity of the outage.

Over the past 3 – 4 years, the upstream business has experienced several forced outages directly tied towards Boiler Feed Pump (BFP) failure with RCR costs averaging ~ $<Retracted due to confidentiality> AUD. To resolve this and reduce the number of forced outages tied to Boiler Feed Pumps, I propose development of a BFP maintenance curve model (1). This model will project the Boiler Feed Pump efficiency curve using actual plant data against the ‘expected’ efficiency of the BFP which is from its initial specifications.

From this, when dips in efficiency are noted, this should be observed and reviewed by the asset engineers to potentially prevent BFP failure from occurring. At the moment, we do not have active Boiler Feed Pump Curve models for monitoring our thermal fleet. Additionally, once our model is built, we can apply this in both a forward and backward looking manner and see if the efficiency % deviation is a strong indicator of BFP failure and if so, adapt our maintenance strategies accordingly.

With increasing competition and declining energy prices due to supply based economics, this will be of interest to the asset engineering team and broader management as it will enable them to 1) Optimise maintenance strategies and 2) Improve profitability through reduction of our financial penalties via improved asset reliability and reduced Forced Outage(s).

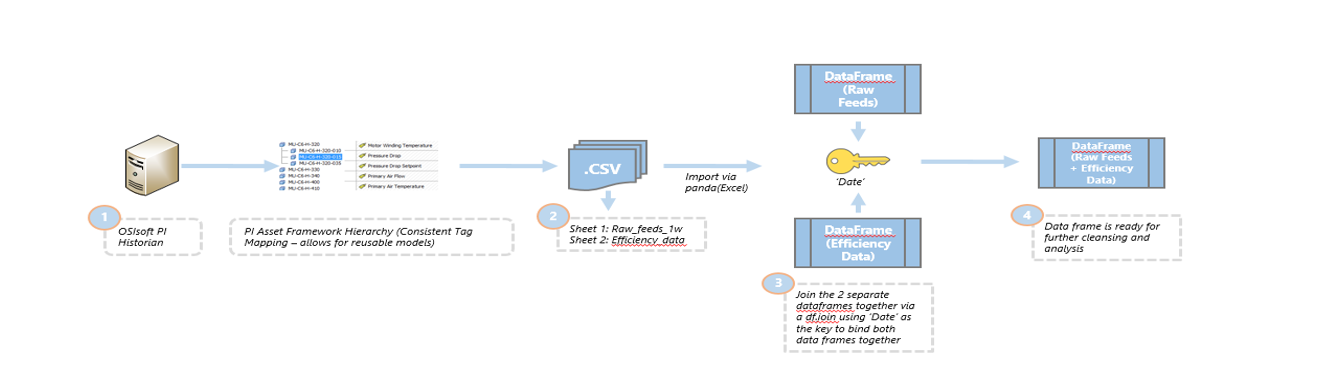
To create the BFP monitoring model, I will need access to the following data repositories:

|  |  |  |  |
| --- | --- | --- | --- |
| Data Required | Source System | Why I need this | Ease of Access (Low, Med, High) |
| Boiler Feed Pump Target Specifications | Technical Documentation | This enables me to create a baseline curve for my efficiency model based off manufacturer specifications. | **Med.**  I will need to review this with the Asset Engineers as the documentation for these pumps was written in the 1970’s. |
| Boiler Feed Pump User Manual | Document Management – Asset Engineering | This will provide me with a better understanding of the BFP operations. This is essential if I am to understand which variables I need to extract data from. | **Low.** I liaise very frequently with the Asset Engineers so this information should be quick to come by and digest. |
| Boiler Feed Pump IoT Sensor Data / Pi Tags | PI Data Historian | PI is our plant historian system which holds information on all of our assets provided the relevant Pi tags have been made. This will be the core system which I will be extracting data form. | **Med.** I will need to liaise with the Pi Specialist to ensure I extract the correct data over the relevant time periods. |

However, as Python has not yet been agreed upon by my work place as a software asset to be used, my approach may need to change. I will assume that I will need to work on this on my personal device instead of my corporate machine. Once I receive permission from the ICT team, I will adjust this accordingly.

1. **Data Transformation** - Once the relevant Pi System Tags for the BFP have been identified and selected with the appropriate time range this will be piped via the Pi Connector System into a .CSV File. Note: *I may need to connect to the maintenance SQL database to extract maintenance data on the Boiler Feed Pumps. I will confirm this and update the above table accordingly.*
2. **Data Import** – The .CSV file will be loaded into my Python Workspace through utilisation of Pandas and the panda.read\_csv function.
3. **Data Cleansing** – There will be perhaps up to 100,000 records (sample size) of readings from the imported CSV. I will need to do a sanity check / data cleanse to check for inconsistencies in the data (i.e. N/A fields or buggy values). Assuming I am bringing in data from another SQL server (i.e. Maintenance history data), there may be a need to interpolate values so the time stamped data matches.
4. **Data Modelling** – Once the data has been suitably cleansed and wrangled for use, I will have to append additional fields to the data frame relating to ‘actual’ and ‘target’ efficiency figure(s) which will be calculated via a efficiency function which will be written in Python. Once these have been appended, I will make use of matplotlib’s plotting functions to plot a model highlighting the discrepancy between the actual and target efficiency values.
5. **Publishing of Model** – Once the model has been developed, I will aim to run this past the asset engineers to determine whether or not they will be able to make informed decisions regarding the maintenance strategy of our Boiler Feed Pumps from analysing the model that has been built in Python.

**Cleansing and Wrangling of Data**



**Figure 1 – Process that was undertaken to extract data required for BFP Analysis**

The cleansing and wrangling of our data followed the process we have outlined above.

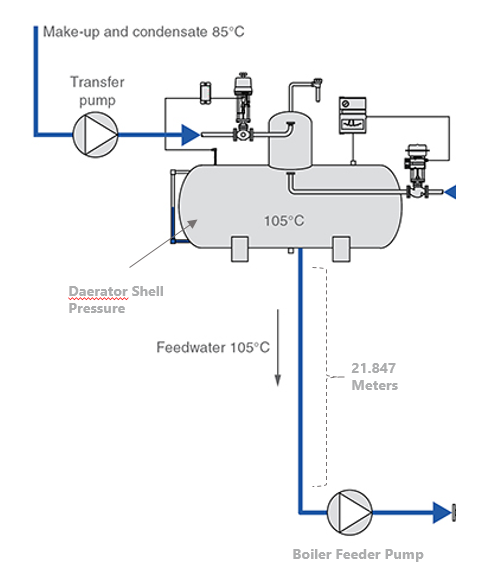
1. **System Data Extraction -** The first step we needed to do was take an extract from our OsiSoft PI System for the relevant tags related to Boiler Feed Pump Activity. This included things such as Vibration, Motor Temperature, Lubrication, Linear Displacement etc. These values were all taken on a specific 5 minute grain which was an linear interpolation of the sampled data over a 5 minute period.
2. **CSVs** – Once the data had been extracted, we placed the information in two ‘tables’. This is because certain information belongs in certain CSVs. While we did not stick to a predefined data model, we believe it is important to maintain separate tables for different types of data.
3. **Joining the data** – Using pd.read\_csv and storing the csv into a dataframe is the first step. The second step was to join these two different data frames using the ‘join’ command which linked the two data frames together by a common index – in our case this is the date.
4. **Review the data** – Inspecting the data revealed that we had almost everything we needed for our Boiler Feed Pump Calculation with the exception of Total Head (Refer to the next slide)

Even with the data joined up, we still lacked certain variables. In our case, one of the key variables required to measure pump degradation is a feature called ‘Total Head’. Total Head is calculated via the below formula where *P*  is the differential pressure between the pumps discharge pressure versus the suction pressure. represents the density of the fluid (water) and *g* represents the constant, gravity.

This can also be written as:

Like most data science projects, we had missing data. However this could not be imputed with regular panda means such as ffill or bfill. In this case, apart from traditional data cleansing (NA’s, values > 0 etc.) we also had to contend with the missing suction pressure. If we miss the suction pressure, we cannot calculate the rate of change in ***P ;*** Hence we would not be able to calculate Total Head.

While we did not have Suction Pressure, we noted that for every Boiler Feed Pump it is connected to the deaerator. The deaerator is a part of plant which strips oxygen from fluids (it de-aerates them). We have a sensor here with recordable deaerator pressure. Further analysis from engineering drawings yields the distance from the deaerator through to the Boiler Feed Pump - **21.847 Meters**.



With this information, we can convert the height using the Standard Instrumentation metrics converting Meters to an appropriate kPa metric. (1 kPa = 0.1019977W Meters).

Combining this with the deaerator pressure sensor enables us to recreate the suction pressure value as shown below.

Additionally, to ensure the values we had were in an acceptable range we included the following filter.

If the filter is to be defined as ‘on’ then:

* Flow rate must be greater than 0
* Pump RPM must be greater than 0
* Total Head must be greater than 0