Data Engineer Task

In this task you will encounter real business cases in Razor Labs Data engineering role which includes:

1. Data Exploration
2. Data pre processing
3. Data pipeline

In your answers, please be specific, get into details and summarize your answers
We are sure you will find this session fun and productive.

1. Data exploration & structuring

At Razor Labs we handle telemetry (i.e, the collection of measurements or other data at remote points) from heavy machinery and production lines around the world. Each record produced by a sensor usually has a sensor name, value and date. Our mission is to analyze the data streams, transform unstructured data to structured data and resample it to a required sampling rate.

Our main goal is to identify the relevant machine and sensor data for each record. This **data is extracted** from the "**name**" column from our "**dataset.csv**" file.

Example Record:

Name, time, value
Hamburg.PLC0640.TX640.D1.VT003, 59:42.4, 3640

The **name** (column) value presented in the example above is defined by these main following parts (**they usually appear as part of the name separated with ./_ in the following order**):

1. **Site name** - The physical location of origin
2. **PLC name** - An identifier for the PLC (Programmable logic controller)
3. **Sub System name** - A specific sub identifier
4. **Reading type**: Input/value/boolean/categorical/logic (see table below for details)
5. **Reading Attribute** - metadata that may or may not be present for each reading

Following columns are **time**, which indicates when the value was recorded, and last is **value**, which represents the actual value.

Telemetry Reading Types (table to assist you in parsing **Reading Type**)

Type Name	Reading type values (multiple options for each value)				
Input (number or boolean)	_I_	I1	I2	I3	D1
Status (True/False)	_V	.V	.COIL_	S13	
Output (number or boolean)	O1	.OUT			
Alarm (True/False)	_A_	.A_	.A.	.ST	.DN
Logic	.TMR	.PID	.EN		

** Please pay attention to the "Telemetry Reading Types" table; it will assist you in the data parsing process. You are not required to understand the meaning of the data just know how to correctly structure it.

I. Create a structured dataset out of the provided csv named "**dataset.csv**". The expected result is an intermediate dataset containing the following columns:
   - **Site name**
   - **PLC name**
   - **Sub System name**
   - **Reading type**
   - **Reading Attribute**
   - **Reading Date**
   - **Value**

***Think about how you handle data that doesn't have all the attributes above present or that is not clear how to parse (no information provided to you).

II. Join the structured dataset with the provided “**metadata.csv**” on the “machine_type” column. In order to distinguish between each machine type use the following table for assistance:

Machine Types (table to assist you in joining metadata.csv with dataset.csv)

Machine Type	**PLC Name**
excavator	PLC303
wheel trencher	PLC0120
generator	PLC0240
front loader	PLC970
scraper	PLC030

2. Data pre-processing

In order to prep the data for an ML model that will predict electrical current you must resample the data. All records with the **Reading Type** of **input** (that are a **number**) must be **resampled to a 5 second frequency** (linear interpolation, see this reference example: https://pythonnumericalmethods.berkeley.edu/notebooks/chapter17.02-Linear-Interpolation.html).

For example:

Hamburg.PLC303.HP703_I_ZT1,27-Oct-2016 11:59:03.788 PM,xxxxxxxx
Hamburg.PLC303.HP703_I_ZT1,27-Oct-2016 11:59:05.884 PM,xxxxxxxx
Hamburg.PLC303.HP703_I_ZT1,27-Oct-2016 11:59:07.004 PM,xxxxxxxx
Hamburg.PLC303.HP703_I_ZT1,27-Oct-2016 11:59:09.369 PM,xxxxxxxx
Hamburg.PLC303.HP703_I_ZT1,27-Oct-2016 11:59:11.892 PM,xxxxxxxx

Will be converted to:

Hamburg.PLC303.HP703_I_ZT1,27-Oct-2016 **11:59:03.000** PM,xxxxxxxx
Hamburg.PLC303.HP703_I_ZT1,27-Oct-2016 **11:59:08.000** PM,xxxxxxxx
Hamburg.PLC303.HP703_I_ZT1,27-Oct-2016 **11:59:013.000** PM,xxxxxxxx
Hamburg.PLC303.HP703_I_ZT1,27-Oct-2016 **11:59:18.000** PM,xxxxxxxx
Hamburg.PLC303.HP703_I_ZT1,27-Oct-2016 **11:59:23.000** PM,xxxxxxxx

** Notice the difference in time between samples, **remember** that the **values** should change accordingly.

3. End to end pipeline

It is time to set up an end to end pipeline. The pipeline will be divided to multiple stages and connected via an orchestrator:

1. Data structuring
2. Loading and preprocess (resampling)
3. Predict (using a mock model)

In order to support the amount of data that is streamed to us every day you are required to write the pipeline in a distributed processing framework such as **spark** (assume that multiple csv's are uploaded simultaneously to a dedicated bucket all day long and require distributed processing for the volume of data generated).

Input:
Telemetry in CSV format like "dataset.csv".

Output:
Structured dataset which also includes resampling of the relevant records

Task Deliverables:

1. Pipeline code in **python**.
2. Intermediate and final datasets and results (files)..
3. Explanation document on the pipeline and the attached results.