XML – JSON 데이터 대조 및 비교 분석

해당 포맷의 파일들은 구조를 하나하나 확인해가며 직접 눈으로 값을 대조하기보다, 간단히 코드를 작성하여 측정 파라미터와 값이 있는 부분을 파싱하여 값을 대조하는 것이 더욱 효율적일 것이라고 생각하였다. 그렇기에, 파이썬을 이용하여 들어오는 파일 포맷에 따라 분기를 나누고, 파일 포맷별로 파라미터와 값이 있는 부분을 분석하여 그 부분만 정확히 파싱해내어 CSV 파일로 내보내는 코드를 작성해서 값 분석에 활용해보았다.

일단, 코드를 작성하기 전에 어떤 부분이 측정값 대조에 필요한 파라미터인지를 원본 XML 파일과 초음파 기기에서 검사 결과가 그대로 저장된 데이터라고 추정하는 XML 파일 두 가지를 대조하여 알아내는 과정을 거쳤다. 원본 파일이 일반적인 XML 형식의 데이터와 조금 달랐기 때문에 일반적인 형태로 정제하여 대조해보았고, 해당 과정은 코드를 사용하기보다 파싱에 필요한 부분을 찾아내는 과정이었기에 눈으로 구조를 따라가며 파싱할 부분과 그럴 필요가 없는 부분을 구분했다.

결론적으로, 원본 XML 파일의 값들은 전체적인 검사 결과 파일(.xml)에서의 ScanMode, ParameterName, DisplayValue가 가리키는 값들을 의미했다. 두 파일은 동일한 검사자의 검사결과 데이터였으므로, 겹치는 값 또한 많았는데, 실제 데이터를 전송할 때 넘겨야 하는 값은 모든 세 가지 값 쌍이 아니라 이 데이터 쌍 중 결과 분석에 필요한 데이터만 정제한 값이라고 추측하였다.

1) XML 파싱
   A. 코드에서는 ScanMode, ParameterName, DisplayValue 이 세가지 값 중심으로 데이터를 파싱한다. 이 세가지 값이 쌍으로 존재하지 않을 경우 CSV 파일로 넘어가지지 않으며 값이 쌍으로 모두 존재할 때만 CSV 파일로 넘어간다. 코드 실행 결과, ScanMode에는 총 4가지 모드가 존재하는 것으로 파악하였고, 각각 MM; MMode, 2D; 2D Mode, CW; 연속파 도플러 모드, PW; 펄스파 도플러 모드인 것으로 추측된다. 코드를 실행하여 얻어낸 CSV 파일과 원본 파일을 대조해본 결과, 실제 데이터 전송을 할 때, MMode와 2D Mode 두가지 모드로 모두 측정 가능한 값은 MMode 값만, 두가지 모드 중 한 가지 모드로만 측정 가능한 값은 해당 모드의 값으로 데이터를 전송하는 것이 아닐까라고 조심스럽게 추측해보았다. 또한 같은 모드로 같은 값을 두 번 이상 측정하는 경우가 있었는데, 이 경우 ResultNo가 -1, 0, 1, 2 순으로 데이터가 저장된 것을 확인하였으며 대체로 이런 경우 ResultNo이 0일 때의 값이 데이터 전송을 할 때 필요한 값이지 않을까 추측하였다.
   B. 해당 규칙은 모드가 도플러 모드일 때도 대체로 적용되고 있는 것처럼 보였는데, 도플러 모드일 때는 규칙이 완전하게 맞아떨어지는 것은 아니라고 판단하였다. 그럼에도 XML 포맷의 초음파 데이터를 분석할 때 해당 규칙을 적용해서 분석해보는 것은 매우 효율적일 것이라고 생각한다.

여기까지가 XML 포맷의 파일을 대조, 비교, 분석한 결과이다. 이 결과를 가지고 제일 데이터 양이 많은 JSON 파일 또한 대조해 보았다.

추가적으로, 첫 주 업무 내용에서 JSON 파일의 말단 파라미터(실제 측정 파라미터와는 관련이 없어보인다.)를 분리해내는 작업을 하고 파라미터들을 살펴보았었다. 그러나 사실상 말단 파라미터는 상대적으로 중요한 값이 아니며, 그 당시에는 실제로 넘어가는 데이터에 어떤 값이 존재하는 지 몰랐기 때문에 실제 측정 파라미터와 값을 구분해 낼 수 없어서 착오가 있었던 것 같다.

결국 실제로 측정값을 나타내는 파라미터는 JSON에서 LabelName과 DoubleValue 또는 RoundedValue가 가리키는 값이라고 분석하였다. XML 파일 형식과 달리 해당 JSON 파일은 꽤 깊은 트리 구조로 되어 있었기 때문에 해당 파라미터들이 어떤 태그를 타고 들어가야 존재하는지를 알아냈어야 했다. 이 과정은 XML 구조 분석과 같이 눈으로 구조를 파악해보며 파라미터가 어디에 위치해 있는지를 파악하고, 해당 위치를 파싱 코드에 적용하였다.

2) JSON 파싱

A. XML과 마찬가지로 측정 파라미터와 그에 해당하는 값만을 파싱한다. 역시나 마찬가지로 이 두 가지 값이 쌍으로 존재할 경우만 CSV 파일에 데이터가 저장된다. 앞서 분석했던 결과를 적용해봤을 때, MCR_AO_DIAM_D_MMODE_CARDIAC와 같이 측정 파라미터 이름/측정 모드가 한 파라미터 이름으로 들어와있었다.

B. 대체로 구조가 LavelName -> FormattedElements -> 값 순이며, Redirected 부분이 null이 아닐 경우 Redirected 부분에도 LavelName -> FormattedElements -> 값 구조가 반복되어서 이 부분을 놓치면 모든 측정 파라미터를 알 수 없다는 점이 까다로웠다.

C. 다만 JSON 포맷의 파일의 데이터 전송 형식을 알 수 없어 정확히 어떤 값이 전송되어야 하는 지는 추측하지 못했다. XML 파일과 겹치는 값들은 간단히 대조해 볼 수 있었지만 겹치는 값이 많지 않아 이 부분에 대해서는 조금 더 정보가 있어야 할 것 같다는 판단이다.