Data Engineer Challenge — XML Attribute Extraction

Context

Cypris ingests structured XML stored in Google Cloud Storage (GCS) that represent patents. Downstream systems depend on extracting specific attributes reliably under imperfect conditions (messy XML, partial data, network blips). In this challenge you will be working with the following representative snippet of this XML:

```
<root>
  ...
  <application-reference ucid="US-XXXXXXXX-A" is-representative="NO" us-art-unit="9999" us-series-code
    <document-id mxw-id="ABCD99999999" load-source="docdb" format="epo">
      <country>US</country>
      <doc-number>999000888</doc-number>
      <kind>A</kind>
      <date>20051213</date>
      <lang>EN</lang>
    </document-id>
    <document-id mxw-id="ABCD88888888" load-source="patent-office" format="original">
      <country>US</country>
      <doc-number>66667777</doc-number>
      <lang>EN</lang>
    </document-id>
    ...
  </application-reference>
  ...
</root>
```

Task

Write a Python program that given a document which contains this snippet, returns all doc-number values (e.g. 999000888, 66667777) in priority order of epo first, then patent-office.

- You must document any and all assumptions you make about the structure of the document.
- Handle errors in a sane and uniform way.

Deliverables

Email a link to a git repository with the code and a README.md that outlines how to run it to prove the output.

Please use uv for and dependency installation/virtual environment handling.