Part 1 Instructions

1. Write a short profile of each file we have given you. Provide a narrative description of the file (50-100 words per file), including its format, properties, contents, and MD5 checksum. This website has a checksum calculator plus gives instructions for running a checksum in Unix and Linux systems: http://sha1md5checksum.bugaco.com/cryptocalc/index.html

- **Profile for File A:**
  - File A contains a list of complaints filed by consumers of various financial institution. Each complaint has events associated with it as well as company information for which the complaint is directed to. It also contains the information about the product type, issue type, how the complaint is submitted, what is the response time for the complaint and an optional consumer narrative specifying the actual feedback from the customer. The file is provided in XML tree format with all the different complaints are presented as child nodes of parent node "ConsumerComplaints". This XML file is equivalent of having a ConsumerComplaints table in the database and each child node "Complaint" represents a row in that table. The XML tree in the file is properly indented with 4 spaces indentation between parent and child nodes.
  - The MD5Sum for this file is: **637737835b3639596bf6db0fa0fff691** (Please note I have used MD5 utility on Mac to calculate this value) and it is 10764 bytes in size.
- **Profile for File B:**
  - File B contains a list of complaints filed by consumers of various financial institution. Each complaint has events associated with it as well as company information for which the complaint is directed to. It also contains the information about the product type, issue type, how the complaint is submitted, what is the response time for the complaint and an optional consumer narrative specifying the actual feedback from the customer. The file is provided in XML tree format with all the different complaints are presented as child nodes of parent node "ConsumerComplaints". This XML file is equivalent of having a ConsumerComplaints table in the database and each child node "Complaint" represents a row in that table. In this sense it is actually same as File A, some of the difference I noticed are that for some of the consumers' sensitive information, File A uses 'XXXX', but File B defined a new entity "redaction" with a constant value of 'XXXX' to hide consumers' sensitive information. Also file B XML is not properly indented hence it is smaller in size with just 9876 bytes in size compared to the file A which 10764 bytes in size. Another difference is that for **response.timely** attribute the value used are **'yes/no'** rather than **'Y/N'**. There are some trailing spaces in attribute values like for **event.type**
  - The MD5Sum for this file is: **c2fb08e9a52dc8cd4d7b0c195061c783** (Please note I have used MD5 utility on Mac to calculate this value)

2. Create a DTD for each XML file. (Note that File B currently contains a minimal internal DTD; you must expand this DTD to fully capture the elements, attributes, etc., in the document.) You may use a DTD generator to generate drafts of your DTDs, but make sure you manually inspect the resulting DTDs to ensure the results are accurate. DTD generators tend to generate attributes, constraints, and default values that do not accurately reflect the possibilities and requirements for your XML documents. Your choices of attributes, constraints, and default values must be justified. You will lose points if we see useless attributes, inaccurate constraints, or unjustified default values in your DTD. Ensure that your XML documents validate against your DTDs.

For DTDs please check the **File-A.xml** & **File-B.xml** which are part of this zip. Both of these files contain there corresponding dtds. Both the XML files are validated against their respective dtds also. For File A: I have added enumerated values for the attributes like **submitted.via**, **response.consumerDisputed**, **response.timely, event.type** in order to maintain the sanctity of the data. I have refrained myself from using the default value for any attribute but just defined the enumerated list of values which can be easily seen and entered by the users, as I was not sure which value should be used as default and analysis for that seems out of scope to me. Although all the complaint IDs are numeric and seems unique but still I have kept the datatype of the attribute as CDATA as there can be some system that might not conform to the constraint of the ID type attribute and analysis of all the downstream/upstream system is out of the scope of the project, so I thought it would be best to keep this attribute as CDATA. For File B in addition to things described above, I have also defined an enumerated list of values for **complaint.submissionType**. Also I was not able to define enumerated values for **event.type** as the values for that attribute have erroneous trailing spaces. The canonicalization process will fix it.

3. Canonicalize the two data files and run checksums again to check for equivalence. Please make sure to document your checksum results (which you will report in step 5). Optionally, you may write and employ a script to conduct the canonicalization process; if you do, please make sure to submit a well-documented, readable text file of the script. For an example of canonicalization, see videos for week 10.

Please check attached files.

4. Create and document the DTD of the final, canonlicalized data file, from step 3 above.

For canonicalize version please check file '**FDC_FinalProject_canonicalize.xml**'

5. In a separate document, answer the following reflection prompts:
a) Describe your process for canonicalization (i.e., decisions, actions, representation selection, attribute issues, provenance decisions). Report the checksum values after canonicalization.

For the canonicalization process I have derived inspiration from https://www.xml.com/pub/a/ws/2002/09/18/c14n.html and https://www.di-mgt.com.au/xmldsig-c14n.html .

Following are the steps followed in the canonicalization process:

1. Removed the XML declaration and DTD.
2. Whitespace outside of the root node are normalized.
3. All the comments (if any) are removed.
4. The file is encoded in UTF-8.
5. Normalized all the line breaks to #xA before starting to process the XML file.
6. All whitespaces in CDATA is retained (excluding the ones that are normalized during the normalization in step 5).
7. Normalized attribute values:
    a. All the attribute values are quotes in double quotes.
    b. Handle special characters
    c. Any leading and trailing spaces are removed
8. Normalized white space between start and end tags
    a. No white space between the left angle bracket ('<') and the name of a start element. Similarly, there should be no space between a slash ('/') and the name of an end element.
    b. A single #x20 character between the element name and the first attribute name, if present.
    c. No white space before and after the equality sign in attribute-value pairs.
    d. A single #x20 character between attribute-value pairs.
    e. No white space following the closing double quote of the last attribute's value.
    f. If there are no attributes, there should be no white space between the element name and the right-angle bracket '>'.
    g. There should be no whitespace between '>' of start tag and CDATA, also between '<' of end tag and CDATA.
9. All elements are converted to start-end tag pairs including the empty elements.
10. All the elements within a parent element are arranged in lexicographical order.
11. All the attributes for an element are also arranged in lexicographical order.
12. In File B, I have removed the **redaction** entity and replaced it with **'XXXX'** in CDATA of "**complaint.consumerNarrative**".
13. Indent all the elements correctly (each child element is 4 spaces indented from its parent element)
    a. Each parent element start/end tag is on its own line followed by a CR (line feed, I have used Unix one).
    b. Each child element start tag will be on the same line as the text of that element and end tag will be on the same line as the end of the text.
14. Normalized all the values for all the Boolean attributes to 'Y/N'. The **response.timely** attribute value in File B are changed as **yes** to **Y** and **no** to **N**

15. Changed submitted element as attribute (similar to file B). This field looks more like a metadata for complaint and I don't see any further information which needs to be captured as part of submitted element, so kept it as an attribute in canonicalize form.
16. Data reconciliation – matched the data from File A and File B for each node and added the attributes data wherever data is missing from either file. Few examples:
    a. For complaint ids **2364257, 837784** in file B, the submission type is missing, so I have added submissionType as "web" (based on the data from File A).
    b. For complaint id **837784, 14038 timely** attribute was missing in file in B. So, added accordingly.

All the changes are done in **SublimeText** by hand and then MD5 utility on Mac is used to calculate the MD5 sum for equivalence.
MD5sum for the final file (both file A and file b final canonicalized version also ends with same MD5sum) is **0c712799b04c5195fc0f90771f976787.**

In order to scale the canonicalization process described above, ideally a script should be written for processing. In adherence to the provenance, the steps mentioned above are documented so that the canonicalization process can be utilized in future. Also, a DTD for canonicalized file is also created.

b) How does the way data is represented impact reproducibility?

In a perfectly reproducible system, any equivalent data undergoing a set of well-defined instructions should produce the same end results irrespective of the format/structure of the representation of the data. For e.g. consider a perfectly reproducible system, with information about our class with all student names, their address, their age & any other information. Any query to fetch information from such system should result in same data irrespective of how the source data is stored (in DB, as XML, as json etc). But in real world, systems expect their data to be in certain format, encoding and structure. And if non-conforming data is sent to the system the results are more likely than not error prone and unreliable. This is the reason why data standardization techniques like XML canonicalization are important. Method of canonicalization (which can be defined as set of custom pre-processing rules) is a way to help ensure that two functionally equivalent datasets when processed through the same system produces same end-results. In the task at hand, we canonicalize (define steps to canonicalize) the XML files, once both the File-A and File-B are canonicalized, both should respond to the same instruction with same results. For e.g. before canonicalization if we query the consumerNarrative for same complaint Id from both File A and File B, both answers though should be same but actually will result in two different strings but after canonicalization both File A and File B will result in same string. Hence, the canonicalization process improves the reproducibility by actually standardizing the data according to the pre-defined set of rules.

c) How may your canonicalization support the overarching goals of data curation (revisit objectives and activities of Week 1)?

- **Reproducibility**
    - As per answer to question B, canonicalization supports reproducibility by ensuring the fact that functionally equivalent data points in either file produces same results when any valid XML query/transformation is performed.
- **Collection/Organization/Storage**
    - The canonicalization is actually a method of organizing the data. The canonicalization pre-processing steps done on the raw data files ensure that the equivalent data at the end of the process will results in functionally uniform and standardized data in the context of XML which adheres to a particular DTD. This process can be part of the 'Extraction' in the ETL pipeline. Once the canonicalization is done, it's easier for the storage system to organize the data. For e.g. store the canonicalized version of XML only and any incremental data update (conforming to DTD) will be done to this canonicalized XML only.
- **Preservation**
    - As part of this project, I have created DTDs for both the files before canonicalization and DTD for canonicalized file also. These DTDs not only serves as validators for XML but also help ensure that the data in XML can be understood in future. MD5 sums are calculated and documented for future validations. Entire canonicalization process is also documented so that canonicalization process can be done in future either for incremental update or in case the data get corrupted.
- **Access/Discoverability/Workflow:**
    - DTDs defined for each of the file should help users in understanding file format, designing their queries and finding relevant information. Though workflow depends on the downstream system, but this process has surely streamlined the data query for the downstream system by making sure that query on equivalent dataset will produce same results.
- **Identification**
    - DTDs defined serves as validators for the data. MD5sum documented should help in ensuring the authenticity of the data. Canonicalization process done above should aid in identification of same data from differently formatted XML files as it generates a functionally equivalent canonical XML when the original XML contains the same data.
- **Integration/Reformatting/Modification**
    - The activities done as part of this project represents more of an act of these data curation activities. The explicit DTDs will help in integration in future. Any XML containing same data when processed through the canonicalization steps should reformats to functionally equivalent canonical XML. Ample documentation is provided on canonicalization process and any other changes made which adhere to the best practices for an act of reformatting. Again, explicit DTDs are provided that will help in data management and correction if needed. For e.g. as part of this project itself, as the last step in my canonicalization process I was able to do some data reconciliation.
- **Provenance**

- Canonicalization done here will ensure that any process done on functionally equivalent data will result in same output which improve the tracking of functionally similar data (even though in different formats, structures). Detailed documentation of each of the steps of canonicalization is also provided to keep track of pre-processing steps done on the raw data. And can be used in future as reference to backtrack any data point.

- **Sharing/Communication**
  - These activities are supported by having more documentation in the form of DTDs and canonicalization process. Also, the differently formatted equivalent data after the canonicalization produces functionally equivalent XML which can be easily distributed. The canonicalization process increases the interoperability, as two different sources can feed data into the system and downstream systems can consume final canonicalized data.

- **Compliance/Security**
  - I don't think canonicalization has done anything to support these activities but explicit DTDs and MD5 sum should aid in supporting compliance objectives and security of the system.

d) Which additional curation activities would you recommend to enhance the data set for future discovery and use?

To enhance discovery and use, I would like to make date as child element of event and may be further split it by year, month and day, which will help users in performing queries like what all complaints are filed against Bank of America in the year 2018. I assume queries like these should be fairly common in systems consuming this data and by providing support for such queries we will definitely give a boost for future use. Also, I would like to split issue type string into separate issueType child elements (when multiple issuetypes are mentioned like for complaint id 759222 a} Loan modification, b} collection, c} foreclosure), which should also help in improving query experience as users can query data for individual issuetype. Also, when we split issue type, it will be easier to enumerate different issueTypes in the system. May be in future we can define an XSD for the XML, so that we can have some validation for values in the element like productType. But with DTD, it might be better to convert the productype and subproduct as an attribute of product and have enumeration, so that for any complaint we will have a correct product type/subproduct. Currently this validation is falling on the consumer systems.

Part 2 Instructions

1. Write a convincing memo (650-word max) explaining why data curation services are important. Assume that the memo is written for your new director, who is not familiar with data curation, and not convinced whether to keep funding this work. You will want to make sure to introduce data curation within the broader context of data science. You will need to

cover the key areas that you think are the most important for data curation at your company. We ask that you incorporate at least two of the following topics into your memo: Provenance, Policy, Metadata, and/or Preservation.

To,
Director, Data Science Group

From,
Pushpit Saxena
Senior Data Scientist

Date: 12/02/2018

Subject: Data curation and its importance for our business process & success

Sir,
As a government agency the foremost and sacred responsibility that is bestowed upon us by this great nation of ours is transparency and effective reporting of the crucial data points that our agency collected and provide insights into that data using the analysis techniques that we are best known for. People of this nation look towards us with a belief that as a government oversight agency we will have their backs. Our complaints department serves as an important oversight body over the financial corporations conducting their business in our nation. And we as an oversight agency have an utmost responsibility to provide accurate reports to the general public so that they can make an informed decision and their trust increases more on our agency and administration.

One of the most important things we do is to provide platform for the users to submit their complaints regarding some of the transactions they have done with any financial institution operating in the country. This database of information is very useful as this provides the overall picture of these institutions and all kinds of problems that are prevailing in the system. The volume of complaints received by our agency are continuously increasing as more and more people have increased trust in our reporting and are feeling empowered to submit their complaints. In order to effectively manage such large volumes of data and process it to provide accurate and informational reports to the public, we have data science as the center pillar of our organization. Data science helps us quickly digest large amount of data and provide insights, which at the end help our editors generate good, timely and trustworthy reports.

Data science is a complex field, but broadly it has two main components: a) Data curation, and b) Data analytics. Data curation is the one that this memo concerns.

According to the field experts, specifically Prof. Allen Renear of UIUC, "Data curation is concerned with all the aspect of data management in order to create a consistent and efficient data pipeline which will help in generating detailed, reliable and reproducible analysis." With no data curation, there is a high risk of decline in the quality of the analysis which at the end will hurt the reputation

and reliability of our agency. Most people are of the thought that, the greatness of a data scientist is wrapped up in building better algorithms. However, data science experts say that building a better algorithm is just like building a super-fast rocket ship. It is good on the assumption that the ship is in the right direction. Nobody is interested in moving faster if they are heading in the wrong direction for analysis but this is what truly happens in many organizations. Data curation plays a crucial role in ensuring that the rocket points in the right direction and have all the fuel in place to propel as fast as it can towards its goal.

Few important things the data curation covers:

1. **Organizing and Understanding the data:** This according to me is one of the most important aspect of data curation. In this age of IOT, we collect data in wide variety of formats. Even when the type of data format (e.g. XML) is same still the same data collected by different sources can have wildly different DTDs (similar issue can occur in data collected as json/db etc. Subtle difference in the DTDs, json structure, db-schema can lead to a very different looking end data, for which our data scientists will have to invest time cleaning up and which might lead to wildly different analysis also). One example came to my mind is our recent switch of the systems for managing complaints system. Even though the data coming from the new system was in XML format and contains same data, there were small differences in the DTDs. With our robust data curation pipeline, we have maintained the detailed documentation of the old DTD and have adhered to our provenance guidelines. This helped us in quickly turned around and switch to new data collection system. For our data scientist the process was quick and painless as they were getting the same end data and all of their previous as well as new models worked perfectly fine. Now, without our data curation pipeline, such kind of system switches will take a lot more man-hours.

2. **Maintaining the existing data:** Data curation also helps in maintaining the existing data in the system, and make the process of incremental data collection seamless as data curation activities focus on defining the policy of data collection and management. It mandates a necessary level of documentation which leads to better understanding and usability in future. It also encompasses preservation of metadata (e.g. source, authors, format etc.) which ensures complete and continued understanding of the data collected. Along with good storage strategy, this will help in preserving the value of data for future use and allow our data scientist to uncover better insights on legacy data as well or to run their improved models seamlessly on the old datasets as well.

3. **Audit records:** Provenance is one of the main activities included in the data curation pipeline. With provenance guidelines system preserve an audit trail of data with information like, the source of the data, different formats of the data, who collected the data, how the data is collected, is there any modification done on raw data (canonicalization, normalization etc.), which dataset is used to build which ML model, what kind of ML models are built on the data, what reports the data is part of. The answer to these questions seems trivial, but the real strength of this can be attributed to the fact that this empowered the agency to easily track down each and every data point/insight that it has published or will publish in future which leads to increased

transparency & stakeholders' confidence at the forefront and also substantially reduce the man-hour spent in digging up this information when questions arises on the agency reports. Also, by maintaining this audit trail of the data, agency can also hope to see improved turnaround for model development and analytics as it will be easier for our data scientists to reuse proven and validated past approaches as stepping stones for new research.

Data curation activities also increase discoverability of the data, streamlines access mechanism, help standardizing workflows around the data, increase interoperability between different departments as with standardized policies and processes it will be easier to share information across, and with central data curation pipeline the agency can implement all the privacy and security checks easily at one place, in fact there are dedicated data curation activities which help in increasing the security of the data systems.

So, to conclude if data curation activities are involved in the procurement, processing and maintenance of the data, our agency will have all the mechanism to support the data that is secure, trustworthy and reusable as well as most of the data scientists can have their valuable time devoted to actual algorithm building and betterment of the machine learning models rather than focusing on improving the data quality in the system. The agency can improve on its commitment to increase transparency and provide the stakeholders with trustworthy and timely reports. As a whole this will increase the trust on the agency which can lead to even better funding for our agency as the law makers will definitely realize the efficient, accurate and detailed oversight that our agency is providing.