A black rectangle with a black background

Description automatically generated with low confidenceLogo, company name

Description automatically generated

DAEN 690

Project Report

Bhavana Macha

Prakruti Rajendra Kothari

Sachin Mote

Sahithi Oddiraju

Saiprem Vemulapalli

Ravinder Singh

Spring 2023

Integrating Data Catalogs with content search technologies

**About the Cover**

Dr. Isaac Gang is an Associate Professor at the George Mason University College of Engineering and Computing, Volgenau School of Engineering, MS Data Analytics Engineering (DAEN) program.

He joined the DAEN faculty in the Fall of 2020 from Texas A&M University-Commerce (TAMUC) where he served as an Assistant Professor of Computer Science as well as the department’s Outreach Coordinator. Before coming to TAMUC, Dr. Gang was an Assistant Professor of Computer Science and Engineering at the University of Mary Hardin-Baylor (UMHB) and an Adjunct Professor of Computer Science at the University of Southern Mississippi’s School of Computing before joining UMHB.

Dr. Gang is a former DOE grant winner, former President and Board Member of the Association of Computer Educators in Texas (ACET), Industry Advisory Board (IAB) Coordinator, and the Director of CS For All.

His current and primary teaching responsibilities at Mason largely involves Data Analytics Engineering graduate courses along with a mix of CS and AIT graduate courses. He is an affiliate faculty member of GMU’s C4I & Cyber Center.

Dr. Gang’s primary research agenda involves Big Data Analytics (emphasis on data bias and data governance), Cyber Security (ransomware, steganography, and cyberbullying), and Image/Signal Processing.

Contents

Table of Contents

[Abstract 5](#_Toc125979012)

[Section 1: Problem Definition 7](#_Toc125979013)

[1.1 Background 7](#_Toc125979014)

[1.2 Problem Space 7](#_Toc125979015)

[1.3 Research 7](#_Toc125979016)

[1.4 Solution Space 8](#_Toc125979017)

[1.5 Project Objectives 8](#_Toc125979018)

[1.6 Primary User Stories 8](#_Toc125979019)

[1.7 Product Vision 9](#_Toc125979020)

[1.7.1 Scenario #1 9](#_Toc125979021)

[1.7.2 Scenario #2 9](#_Toc125979022)

[Section 2: Datasets 9](#_Toc125979023)

[2.1 Overview 9](#_Toc125979024)

[2.2 Field Descriptions 9](#_Toc125979025)

[2.3 Data Context 10](#_Toc125979026)

[2.4 Data Conditioning 11](#_Toc125979027)

[2.5 Data Quality Assessment 11](#_Toc125979028)

[2.6 Other Data Sources 11](#_Toc125979029)

[2.7 Storage Medium 11](#_Toc125979030)

[2.8 Storage Security 11](#_Toc125979031)

[2.9 Storage Costs 12](#_Toc125979032)

[Section 3: Algorithms & Analysis / ML Model Exploration & Selection 12](#_Toc125979033)

[3.1 Solution Approach 12](#_Toc125979034)

[3.1.1 Systems Architecture 12](#_Toc125979035)

[3.1.2 Systems Security 12](#_Toc125979036)

[3.1.3 Systems Data Flows 12](#_Toc125979037)

[3.1.4 Algorithms & Analysis 12](#_Toc125979038)

[3.2 Machine Learning 12](#_Toc125979039)

[3.2.1 Model Exploration 13](#_Toc125979040)

[3.2.2 Model Selection 13](#_Toc125979041)

[Section 4: Visualizations / ML Model Training, Evaluation, & Validation 13](#_Toc125979042)

[4.1 Overview 13](#_Toc125979043)

[4.2 Visualizations 13](#_Toc125979044)

[4.3 Machine Learning 13](#_Toc125979045)

[4.3.1 Model Training 13](#_Toc125979046)

[4.3.2 Model Evaluation 14](#_Toc125979047)

[4.3.3 Model Validation 14](#_Toc125979048)

[Section 5: Findings 14](#_Toc125979049)

[Section 6: Summary 14](#_Toc125979050)

[Section 7: Future Work 14](#_Toc125979051)

[Appendix A: Glossary 15](#_Toc125979052)

[Appendix B: GitHub Repository 16](#_Toc125979053)

[Overview 16](#_Toc125979054)

[GitHub Repository Link 16](#_Toc125979055)

[GitHub Repository Contents 16](#_Toc125979056)

[Appendix C: Risks 17](#_Toc125979057)

[Sprint 1 Risks 17](#_Toc125979058)

[Sprint 2 Risks 17](#_Toc125979059)

[Sprint 3 Risks 17](#_Toc125979060)

[Sprint 4 Risks 17](#_Toc125979061)

[Sprint 5 Risks 18](#_Toc125979062)

[Appendix D: Agile Development 19](#_Toc125979063)

[Scrum Methodology 19](#_Toc125979064)

[Sprint 1 Analysis 19](#_Toc125979065)

[Sprint 2 Analyis 19](#_Toc125979066)

[Sprint 3 Analysis 20](#_Toc125979067)

[Sprint 4 Analysis 20](#_Toc125979068)

[Sprint 5 Analysis 20](#_Toc125979069)

[Works Cited 21](#_Toc125979070)

**No table of figures entries found.**

**This page intentionally left blank**

Abstract

Abstract

**INSTRUCTIONS**

[NOTE: The project abstract is a separately graded assignment in the course. The final approved project abstract is to be copied word-for-word from the other assignment into this report.]

Write one paragraph of no more than 300 words that summarizes your project. Here are the typical kinds of information found in most abstracts which you should use as an outline as you develop your abstract.

1. The context or background information for your research; the general topic under study; the specific topic of your research.
2. The central questions or statement of the problem your research addresses.
3. What’s already known about this question, what previous research was conducted or shown.
4. The main reason(s), the exigency, the rationale, the goals for your research — why is it important to address these questions? Are you, for example, examining a new topic? Why is that topic worth examining? Are you filling a gap in previous research? Applying new methods to take a fresh look at existing ideas or data? Resolving a dispute within the literature in your field?
5. Your research and/or analytical methods.
6. Your main findings, results, or arguments.
7. The significance or implications of your findings or arguments.

Your abstract should be intelligible on its own, without a reader’s having to read your entire paper.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

**This page intentionally left blank**

Report

# Problem Definition

## Background

In the era of big data and self-service business intelligence, data catalogs have established themselves as the industry standard for managing metadata. The metadata we require today is bigger than the metadata from the BI era. To enlighten those who work with data, a data catalog first concentrates on datasets (the inventory of readily accessible data). The essential capacity of cataloging data—collecting the metadata that identifies and characterizes the inventory of shareable data—underlies all the features and functionalities that a modern data catalog provides. Cataloging cannot be attempted manually due to practical reasons. The initial catalog development and continuing discovery of new datasets both require automated dataset discovery. To maximize the benefits of automation and reduce manual work, it is crucial to use AI (Artificial Intelligence) and machine learning for metadata collecting, semantic inference, and tagging.

A data catalog is a collection of metadata, along with data management and search tools, that aids analysts and other data users in finding the data that they require. Data catalogs also act as a list of the data that is currently available and offer information to assess the suitability of the data for which it is being used. Data catalogs have successfully established themselves as vital to modern data management. Data analysis speed and quality, as well as employee engagement and enthusiasm, change dramatically in organizations where data catalog implementations are successful. The popularity of data catalogs has continuously increased since 2016 and they are deemed to be “the new black in data management and analytics” [1] according to Gartner [1]. The Data Catalog offers several methods to assist you in finding your useful data resources. One can:

* Use both the data resources' actual data and their metadata to conduct a search using pertinent phrases in your catalog.
* Browse the categories you have created for your data resources.
* Get automated data asset suggestions based on your interactions.
* Examine whether data assets are crucial to your company's other users.
* Explore the perspectives associated with a view that is important to you.

Without a catalog, analysts search for data by reading documents, speaking with coworkers, relying on lore, or just using well-known datasets because they are familiar with them. Trial and error, wastage, rework, and repetitive dataset searching are all risks in the process, which frequently forces workers to use "near enough" data while time is running out.

To help users discover the data they need, data catalogs frequently integrate with content search tools. Users may now search for data assets based on a variety of criteria, including data type, source system, data quality, and more, thanks to this connection. This lowers the risk of improper data consumption and enables firms to be sure that the appropriate data is being used for certain business needs. Users can search for data assets in a way that is like how they would look for information on the internet thanks to the integration of a data catalog with content search capabilities. Users can search for specific data assets based on keywords, data attributes, and metadata information thanks to the data catalog indexing engine, which builds a searchable index of all the metadata and data assets within the catalog.

One of the main advantages of this integration is that it makes it easier for businesses to deal with the problem of data silos. A data silo is an isolated group of data that is inaccessible to other organizational units. Organizations can eliminate these data silos and increase the discoverability and accessibility of their data assets by integrating a data catalog with content search capabilities. This aids businesses in maximizing the value of their data assets, preventing data duplication, and lowering the danger of data loss. Fig 1[6].  shows how ​​analysis procedures change with and without data catalogs.

Graphical user interface

Description automatically generated with medium confidence

Fig 1.  Process with and without Data Catalogs

Numerous other features and functions are supported by the data catalog's robust metadata, the most crucial of which are:

* **Data Searching:** Powerful search options include searching by facets, keywords, and     business terms. For non-technical users, natural language search capabilities are especially useful. Particularly helpful and advantageous features include the ranking of search results by relevance and frequency of use [6].
* **Data Access:** With the catalog understanding access protocols and either directly providing access or interacting with access technologies, the user experience from search to evaluation to data access should be seamless. Access protections for sensitive data related to security, privacy, and compliance are included in data access functions [6].
* **Data Evaluation:** It is important to be able to assess dataset’s suitability for analysis use cases without having to download or acquire data first. The ability to preview a dataset, view all associated metadata, view user ratings, read user reviews and curator annotations, and view data quality information are all crucial evaluation features [6].

Including a data catalog with content search capabilities aids companies in improving data governance and increasing the discoverability and accessibility of data assets. The process of overseeing and guaranteeing the reliability, accessibility, and security of data assets is known as data governance. Organizations can quickly find and manage their data assets by combining a data catalog with content search capabilities, ensuring that they are used consistently and in compliance.

A system's capacity to search through data assets like documents, photos, and other sorts of content and provide pertinent results based on a set of search parameters is referred to as having content search capabilities. Regardless of the size or complexity of the data repository, these capabilities are intended to assist users in finding the information they require quickly and simply.

Metadata is data about data and Open Metadata .org is an open-source data catalog tool that supports advanced features like data discovery enabled by keyword search, data associations, complex queries which enable search across tables, topics, dashboards, pipelines, and services. It supports detailed metadata for assets and their components including support for data types, arrays and structs.  Open Metadata also has the feature discover data through association, which allows data discovery based on data lineage, advanced search on metadata operators and Boolean operators, activity feeds, webhooks and slack integration, descriptive metadata which allows to tag to metadata which enables advanced filtering to discover data in the organization. Metadata can refer to a wide range of details that describe the content, context, and structure of a piece of digital data. Some common types of metadata include:

* Descriptive metadata: This kind of metadata offers details about a resource's content, such as the title, author, creation date, and keywords.
* Metadata that describes a resource's structure, such as the number of pages in a document, the chapters in a book, or the sections in a database, is called structural metadata.
* Administrative metadata: This category of metadata offers details about the ownership, copyright status, and access limitations associated with the use and management of a resource.
* Technical metadata: This category of metadata contains details about a resource's format, size, and resolution.
* Preservation metadata: Metadata about resource management and preservation, including its provenance, authenticity, and integrity, is provided by this type of metadata.

Each type of metadata is crucial for describing and classifying digital resources, as well as for enabling their proper administration, preservation, and long-term use.

## Problem Space

Only a few of the ways that search adds value have been examined and quantified in most of the literature to date: by saving time, by improving price transparency, and by increasing awareness.

The files and tables that data workers must locate, and access are known as datasets. They could be kept in a master data repository, data lake, warehouse, or any other shared data resource. Data workers such as consumers, curators, stewards, subject matter experts, etc. are all described by people's metadata. To assist users in finding data, search metadata includes tagging and keywords. The application of transformations and derivations as data is managed throughout its lifecycle is described by processing metadata. Supplier metadata, which provides information about sources and subscription or licensing restrictions, is particularly crucial for data obtained from external sources.

In this project, Elastic search which is built on Apache Lucene will serve as the search engine. It has features like ranked searching, strong query types like phrase queries, wild card queries, proximity queries, fielded searching, and nearest neighbors for high dimensionality vectors. It also has sorting by any field, multiple indices searching with merged results, simultaneous update and searching, flexible faceting, highlighting, joins and result grouping, fast, memory-efficient, typo talent suggestions, and a configurable storage engine.

Open Metadata also has the feature discover data through association, which allows data discovery based on data lineage, advanced search on metadata operators and Boolean operators, activity feeds, webhooks and slack integration, descriptive metadata which allows to tag to metadata which enables advanced filtering to discover data in the organization.

The main idea of the problem is to integrate the capabilities of Apache Lucene with the capabilities of metadata which permits the user to find out best fit of tags and ranking of tags using machine learning to auto generate tags, which allows to populate and track the tags.

|  |
| --- |
|  |

## Research

The Alation data catalog, Informatica corporate data catalog, and Oracle cloud infrastructure data catalog are just a few of the commercial data catalog solutions that have been developed recently [1, 2]. However, there is little study on data catalogs and, to the best of our knowledge, no other thorough literature review, despite an important conversation among practitioners and several commercial products. In 2020, Labadie et al. [3] emphasize the necessity for additional data catalog study, particularly about deployment. Both the commercial and academic worlds have emphasized the need for enhanced data documentation. The FAIR principles [4], which state that data should be findable, accessible, interoperable, and reusable, were first published in a 2016 paper co-authored by more than fifty researchers. They "describe distinct considerations for contemporary data publishing environments with respect to supporting both manual and automated deposition, exploration, sharing, and reuse." A collection of requirements for data-related data defines each concept.

Finding the correct data has gotten harder than ever before since there is more data available than ever before. The GDPR is simply one of many laws and rules that are in place at the same time as more laws and regulations than ever before. Data access is thus becoming more difficult, and data governance is also becoming more difficult. Understanding the type of data, you now have, who is moving it, what is being done with it, and how it needs to be protected is crucial. Data is useless if it is too difficult to utilize, thus you must also be careful not to surround it with too many layers and wrappers. Unfortunately, acquiring and finding the appropriate data can be difficult [5].

Numerous avenues have been explored by research in this field. These have typically concentrated on either improving the search for data in specific areas by developing more domain-specific metadata. For instance, the Research Data Alliance (<https://rd-alliance.org/>) For scientific research data has established several interest groups to investigate the creation of metadata standards for scientific fields as well as a more general working group to create metadata standards catalog to point people to these. Many intergovernmental panels and/or research groups inside governments are also looking into the most effective ways to set up data catalogs for improved search and exploration. Other initiatives have investigated how to extract metadata from web page structured data or data descriptions. A different strategy is the thoughtful design of URL (Uniform Resource Locator) naming systems, which makes it easier to find and index catalogs (hierarchically) based on themes, organizations, etc. (For an example, see <https://insidegovuk.blog.gov.uk/url-standards-for-gov-uk/> for the standards of the United Kingdom.)[7].

**1.3.1 Open Metadata**

Metadata is machine processable data that describes resources, digital or non-digital [12]. According to the literature from various fields, the term "metadata interoperability" has an extremely broad definition and involves several issues that must be solved. On a more fundamental level, machines need to communicate to access and exchange metadata. A machine must have the technical ability to process the metadata information objects that it receives from another. We also need to make sure that metadata is correctly interpreted by both machines and people on an extremely high semantic level.

Organizations are retaliating by employing more personnel and tools to address data-related issues. Teams working on data that are spread out globally are cut off from one another. Data has a "too many tools" problem; these tools do not communicate with one another, have fragmented metadata, and further alienate users. The most crucial component of data lacking today is seamless cooperation around consolidated metadata. Data Collaboration is the core component of Open Metadata because of this. In prior editions, we provided Conversation threads and Activity Feeds to encourage collaboration. Tasks are automatically created and given to data owners during these processes. All these exchanges take place within Open Metadata through micro-workflows and interactions, avoiding interruptions to user workflow and context switching between tools. The 0.11 Release's focus has been data collaboration, for which the foundation has been set in previous versions. Activity Feeds, Conversation Threads, and the option to request descriptions were all added with the 0.9 version. Table, pipeline, dashboard, and subject descriptions can form the basis for tasks. Users have the option of adjusting an existing description or suggesting a new one. The request submission automatically generates a task for the data asset owner. These duties can then be given to the appropriate user. To take part in this activity, other users can publish a response, a comment, or emoji reactions to dialogues. On the User's profile page, you may keep track of every assignment that has been given to a user. On the dataset details page, tasks related to a specific data asset are tracked. Tasks are automatically ended when owners offer descriptions or accept/reject proposals. Without hopping between tools as is now done in many businesses were asking for a description is done over email, Slack, Jira issues, GitHub tasks, etc., all of this is conveniently completed within Open Metadata. The ability to suggest tasks excites us the most. As a result, an organization can crowdsource information and continuously enhance its description, taking collaboration to a new level [8].

In general, the main mechanism to disseminate this availability of data has been the deployment of Open Data catalogs exposing metadata of these datasets, which are easily indexed by general web search engines or specialized dataset search engines like Google Dataset Search [16]. Furthermore, to enable the federation of contents, these Open Data catalogs must consume and be harvested from other catalogs with minimal technical agreements. The metadata schema used in these catalogs is one of these minimal agreements. In the context of Open Data, DCAT is the "de facto" metadata standard. The W3C's Data Catalog vocabulary, a W3C recommendation for describing open data, is known by the abbreviation DCAT [17]. Using a common model and vocabulary that makes it easier to consume and aggregate metadata from various catalogs, DCAT enables a publisher to describe datasets and data services in a catalog. This may make datasets and data services easier to find. Additionally, it enables federated search for datasets across catalogs in various sites using the same query mechanism and structure. It also enables a decentralized approach to publishing

**1.3.2 Elastic Search**

Elasticsearch is an open-source distributed database system capable of real-time full-text search and analytics. It is built in Java based on the Apache Lucene library [13]. According to the book "Elasticsearch Server," it was first released in 2010, and over the years it has gained popularity, currently being "widely used in many common or lesser-known search and data analysis platforms” [14]. Elasticsearch was created to handle big data and is document driven. It can scale to hundreds of servers and stores data as JSON documents. A document's fields are all automatically indexed and searchable. Elasticsearch can perform search in close to real time because it by default performs an index refresh every second. Elasticsearch offers fast full-text search through the inverted index paradigm used in Lucene [15]. Elasticsearch excels at full-text search because it is based on Lucene. Elasticsearch is also a "near real-time" search platform, meaning that the time it takes for a document to get indexed to searchable takes only about one second on average. Elasticsearch is hence well suited for time-sensitive use cases like infrastructure monitoring and security analytics [9].

## Solution Space

Our system aims to deliver an integration which allows to integrate the capabilities offered by

search engine apache Lucene and data catalog tool open metadata which enables use features like best fit tags and ranking of tags.

## Project Objectives

* Objective 1: Determine Best Fit Tags.
  + Search Engine feed metadata tags
  + Data Catalog feed metadata tags
  + Determine “best” descriptors for data resource
* Objective 2: Integrate Search Engine with Data Catalog.
  + Determine where Search and Catalogs (metadata) can seamlessly operate together
* Objective 3: Data Catalog provides “has access” indicators.
  + Mimic a data catalog (momentum) & access rule (ABAC, RBAC, XACML)
  + Execute Access Rules real time

## Primary User Stories

Integrating data catalogs with content search technologies can provide organizations with a powerful tool for discovering, understanding, and using their data. It can improve the search functionality of data catalogs and help users find the information they need quickly and accurately. Another benefit of integrating data catalogs with content search technologies is that it can improve the accuracy of search results. The metadata in a data catalog can be used to disambiguate terms, understand the context of a search, and provide more relevant results. To reach our motive to help the users with all this benefits we are thinking to add following features:

1. Incorporating faceted search Feature – it is a search interface that allows users to filter results based on multiple criteria, also known as facets, such as categories, tags, date ranges, and others. This type of search can help in integrating data catalogs with content search technologies by providing users with a more intuitive and flexible way of accessing and discovering information.[10]

* **How faceted search will benefit:**

By incorporating facets into a search interface, users can quickly narrow down the results to a specific subset that meets their criteria, reducing the amount of time they spend searching and increasing the chances that they'll find what they're looking for. The integration of data catalogs and content search technologies can also provide a more complete view of the information that is available, making it easier for users to understand the context of the information and how it relates to other data sets.

Additionally, the use of facets can help improve the accuracy and relevance of search results, since it allows users to refine their searches based on specific criteria. This can also lead to a better understanding of user behavior and preferences, which can be used to further improve the search experience.

1. **Incorporating Search suggestions Feature -** Search suggestions can help integrate data catalogs with content search technologies by providing users with a more intuitive and efficient search experience. When a user starts typing a query into a search bar, search suggestions can appear in real-time and provide relevant terms, phrases, or questions related to the user's search. This can save users time and effort in finding the information they need.[11]

* **How Search suggestions Feature will benefit:**

The search suggestions can be based on the information contained within the data catalog. This can help ensure that the search suggestions are relevant to the user's needs and can lead them to the right information. The integration of the data catalog with content search technologies can also provide better search results by combining structured data from the catalog with unstructured content.

Also, search suggestions can help improve the data catalog quality by providing feedback on what users are searching for. This can help organizations to identify gaps in their data catalogs and make updates to ensure that the catalog provides the information that users need.

## Product Vision

**For:** MITRE Corporation

**Who:** Employees working in MITRE Corporation.

**Our Product:** Data Catalog Integrated with Content Search Engine Technology

### Scenario #1

A new employee can use data catalog to search and access limited documents and webpages related to new hire and onboarding instead of accessing all the high-level content of the company.

### Scenario #2

A database administrator in a company might use a combination of Search Engine that has results produced from an integrated data catalog to examine and can ensure the data governance.

# Datasets

## Overview

This section briefly discusses the dataset that we used and where we acquired the dataset from. Our Primary dataset is Synthea Covid 19 Dataset. The dataset is in text format.

|  |  |  |  |
| --- | --- | --- | --- |
| **Dataset Title** | **Source** | **Size** | **Type** |
| MITRE Synthetic Health Data | MITRE | 21 GB | Text |

MITRE Synthetic Health Data is an open-source patient population simulation made accessible by The MITRE Corporation, is the source of the data stored within Synthetic Mass. This dataset consists of ten thousand synthetic patient’s records based out in Massachusetts with COVID-19 in the CSV format.[18]. The Synthea COVID-19 data set is a longitudinal set of synthetic COVID-19 patients and their EHR records. This data inventory has different files that contain patient personal information, allergies, immunizations, payers, care plans, observations etc.

## Field Descriptions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Title | URL | Authors | Publication Date: | Type | Tags |
| MITRE Synthetic Health Data | <https://synthea.mitre.org/downloads> | MITRE | 11/01/2020 | Text | “Health", "Covid” |

We used 5 csv files from the MITRE Synthetic Covid 100k health dataset. The “Patient.csv”, “Observations.csv”, “Encounter.csv”, “Procedures.csv” and “Careplans.csv” were merged into one csv file with around 49 columns.

**2.2.1 Patient dataset**

The Patients dataset is a list of patients, as well as their healthcare expenses and coverage. Each patient will have a row with their personal information. It has 25 fields, with the patient ID displaying the unique as stated. This collection contains PII data for patients in the form of string, float 64, and integer data types.

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Data Type** |
| ID | This is the unique patient ID that is used to identify patients and patient information. This ID is initially generated when a patient is admitted, or the record is made. | String |
| BIRTHDATE | This is the birth date of the patient. The format is mm/dd/yyyy. | Date |
| SSN | This is the social security number of the patient. This field does not have null. | Number |
| DRIVERS | This is the driver's license of the patient. It is a specific identification number assigned to a driver by the issuing government agency. It has 9 characters. | String |
| PASSPORT | This is the passport number of the patient. It is a unique identification number, and it has null values for the patient who does not have a passport number. | String |
| FIRST | This is the first name of the patient. | String |
| LAST | This is the last name of the patient. | String |
| MARITAL | This is the marital status of the patient. It is ‘M’ for married, ‘S’ for Single. It has blank fields for patients whose marital status is unknown. | Character |
| RACE | This is the patient’s race. It could be Black, White, Asian, Native etc. | String |
| ETHNICITY | This is the patient’s ethnicity. This indicates if the patient is Hispanic or non-Hispanic. This field does not have null values. | String |
| GENDER | This field indicates the patient’s gender. This field either has ‘M’ for Male or ‘F’ for Female. | Character |
| BIRTHPLACE | This field has the patient’s birthplace. | String |
| ADDRESS | This field has the exact place where the patients are living. | String |
| CITY | This field has city where patient is living | String |
| STATE | This field is the state where the patients are living. | String |
| COUNTY | This field is the County where the patients are living. | String |
| ZIP | This field is the ZIP code (postal code) for the place where the patients are living. | String |
| LAT | This field is the longitude of the place where the patients are living. | Float64 |
| LON | This field is the latitude of the place where the patients are living. | Float64 |
| HEALTHCARE\_EXPENSES | This field contains the total expenses of the individual patient in dollars. | Float64 |

**2.2.2 Procedure dataset**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Data Type** |
| DATE | This field is the start date of the procedure for the patient | Date |
| PATIENT | This is the unique patient ID that is used to identify patients | String |
| ENCOUNTER | This field is a unique ID for the encounter | String |
| CODE | This field gives the code for the patient. | Number |
| DESCRIPTION | This field gives the description about the patient's condition | String |
| BASE\_COST | This field gives base cost for the patient’s procedure. | Number |
| REASONCODE | This field gives resaoncode for the patient’s procedure. | Number |
| REASONDESCRIPTION | This field gives description about the reason. | String |

**2.2.3 CarePlan's dataset**

Care Plan dataset gives details about the patient’s care Plan with its start and end date.

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Data Type** |
| ID | This is the unique patient ID that is used to identify patients and patient information. | String |
| START | This is the Start date of the patient’s CarePlan | Date |
| STOP | This is the Stop date of the patient’s CarePlan | Number |
| PATIENT | This is the unique patient ID that is used to identify patients | String |
| ENCOUNTER | This field is a unique ID for the encounter It can be null | String |
| CODE | This field gives the code for the CarePlan | Number |
| DESCRIPTION | This field gives the description about the patient’s CarePlan | String |

**2.2.4 Observation's dataset**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Data Type** |
| DATE | This field gives the date of the observation. | Date |
| PATIENT | This is the unique patient ID that is used to identify patients | String |
| ENCOUNTER | This field is a unique ID for the encounter It can be null | String |
| CODE | This field gives the code for the Patient | Number |
| DESCRIPTION | This field gives the description about the patient’s Observation | String |

**2.2.5 Encounter's dataset**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Data Type** |
| ID | This is the unique patient ID that is used to identify patient's encounters. | String |
| START | This is the Start date of the patient’s Encounter. | Date |
| STOP | This is the Stop date of the patient’s Encounter. | Date |
| PATIENT | This is the unique patient ID that is used to identify patients. | String |
| ORGANIZATION | This field gives ID about the organization | String |
| PROVIDER | This field gives the code for the Patient’s encounter provider | String |
| PAYER | This field gives the ID of the Payer | String |
| ENCOUNTERCLASS | This field gives the description about the Encounters class | String |
| CODE | This field gives the code for the encounter | Number |
| DESCRIPTION | This field gives the details about the encounter’s description | String |
| BASE\_ENCOUNTER\_COST | This field gives the cost about the encounters. | Number |
| TOTAL\_CLAIM\_COST | This field gives the total claim cost of the encounter | Number |
| PAYER\_COVERAGE | This field gives the cost covered by the payer | Number |
| REASONCODE | This field gives resaoncode for the encounter. | Number |
| REASONDESCRIPTION | This field gives description about the reason. | String |

## Data Context

The main idea of the project is to integrate the best fit tags, tracking tags, scalability and security features of the search engine elastic search to the data catalog open metadata. In this project the primary datasets that we used are MITRE Synthetic Health Data, the dataset is sourced from MITRE.

The team has decided to use the above dataset for two primary reasons. Since the questions requested for a search engine are largely text-based, the above datasets are more relevant for this project. The features are text based, which are like search engine queries, and they allow us to validate the best fit tags and tracking tags for the dataset. To make the dataset searchable, it would need to be indexed and labeled with metadata. Adding categories depending on the themes covered in each document, as well as metadata such as the author, publication date, and source of the text excerpt, might be included. Lastly, using a search engine such as Elasticsearch, the dataset might be integrated into a data catalog. This would allow users to search the dataset for relevant documents based on keywords, tags, or other metadata.

## Data Conditioning

Our Primary dataset Synthea Covid 19 Dataset which is a non-realistic but a synthetic data repository from Synthea (MITRE) that has been generated and maintained as an open source. The repository contains data that has been maintained in various files based on their usage such as patients, allergies, medications, and observations etc. For project, the data gathered must be refined and processed in a way that can be used better with data catalog and search engine.

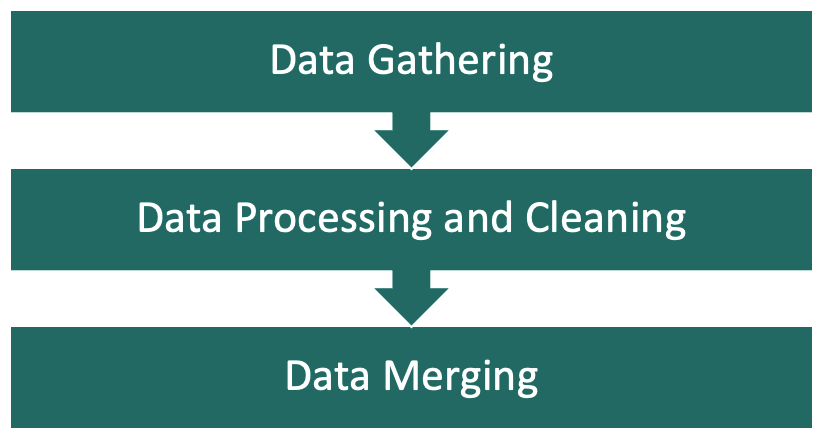


Fig 2.  Data Conditioning Process for Synthea COVID-19 Dataset

From a total of 16 datasets from the repository having more than 100k records, the team has shortlisted a few datasets which are Encounters, Organizations, Patients and Conditions and planned to work further on these datasets to form a single dataset which has a combination of records from all the 4 chosen datasets. The team planned to make use of PySpark and SQL in Databricks to handle these files and perform join Operations to get a final dataset which will be used for tasks and operations that are performed in the future. While considering the idea of creating a new dataset with subsets of data, the team had assumed and identified to work on some features in data such as COVID-19 as a health condition, information of patients who were diagnosed with COVID-19 and organizations where the patients were admitted or treated. The new dataset with identified features will be used to feed the data catalog, generate tags and work on accesses part of this project.

Data Conditioning for BBC News dataset

The BBC news dataset is an open-source self-updating dataset. The dataset is updated automatically through a kernel that functions with fixed frequency and the output of the kernel updates the dataset. The RSS news feed is collected from the BBC news official news site. We are going to be using this dataset for integration with the elastic search engine. This is a single dataset of 4.41 MB. Since this is a small dataset there is not much data conditioning required. The dataset does not have any missing values or skewed values therefore it is good enough for achieving the project objective. Usingo this dataset, the team aims to generate metadata tags that determine the sentiment of the news. We are going to achieve generating tags using OpenMetadata and integrate them with the search engine (elastic search).

## Data Quality Assessment

### Completeness:

The first data set in consideration for this project is Covid 100k Synthetic data generated by MITRE Organization. This data set contains the data in csv files related to Procedure, Observations, Care plans, Encounter and Patient The measure of Completeness was manually calculated from the merged dataset and except for the patient ID column all the columns had some missing values. For this project we are considering using the data sets conditions, encounters. In the dataset file conditions, we are filtering out data based on the code 840539006 as it is the data related to the covid file. There are no null values available in the rest of the fields. The encounters data set is not complete as it has values missing in several fields in reason code and reason description. The second dataset that we considered was BBC News dataset. This dataset is complete and doesn’t have any null values.

### Missing Data:

In the MITRE merged dataset only Patient ID column does not have any missing data rest all the columns have some missing value. The count of missing value for each column is shown in Fig 3. There was no missing data observed in the datasets BBC News.

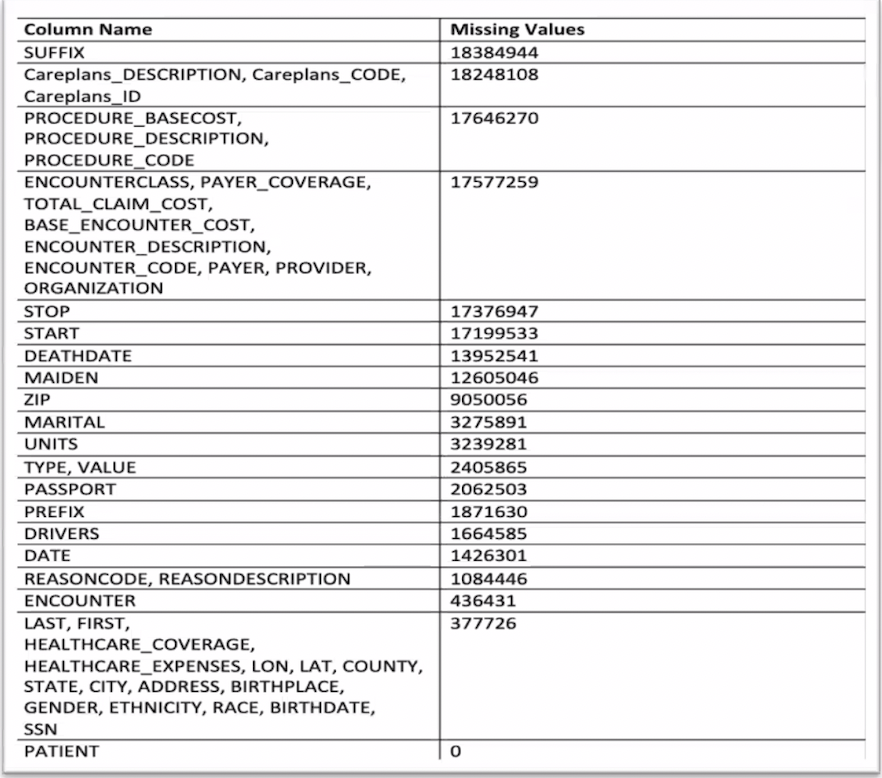


Fig 3. Missing value count for MITRE dataset.

### Uniqueness:

For the MITRE Dataset, many columns in the Merged dataset have numerous unique values. On the other hand, other columns, including ENCOUNTERCLASS, RACE, SUFFIX, PREFIX, ETHNICITY, GENDER, BASE ENCOUNTER COST, TYPE, TOTAL CLAIM COST, MARITAL, and STATE, contain a small number of distinct values, suggesting that some of them represent discrete categorical variables.

The number of unique values for the BBC news is as follows. The tile field contains 13539 values. Pub Date stands for publishing date which contains the date values between 5Sep17 and 25Feb23. The field guide contains 12947 unique values. The RSS Feed description contains 13321 unique values. There are several repeated values for patient, providers, code, description in the encounters data set as the medical data is recorded multiple times. However, the Id column is unique. In the conditions data set the data set is unique and there are no repeated values.

### Accuracy:

Most of the data seems to be accurate except for the field guide in BBC news which is URL provided for the specific article. Some articles might be missing because of the extensive timeline of 5 years from 2017 to 2023. As the data generated for the covid data is synthetic, the data sets encounters and covid data set is accurate.

### Atomicity:

The level of atomicity is very high in the MITRE Merged dataset because most of its columns cannot be further decomposed into smaller attributes and the atomicity of the data is assessed as high for and BBC news data set as there are no fields missing. .

### Conformity:

Most of the dataset's properties have uniform formatting, ensuring the data's consistency and standardization. However, a few columns have data in text and numbers which needs to be processed.

### Overall Quality:

The Overall quality of datasets BBC news is good as they do not have much missing values and most values provided are within the context of data. The overall data quality of the dataset conditions and encounters is moderate as some of the columns have missing values and the missing reason has not been justified.

## Other Data Sources

**2.6.1 BBC News Dataset**

BBC News dataset is available on Kaggle which is a collection of 2,225 news articles published by the BBC between 2004 and 2005. The dataset includes news articles in five distinct categories, including business, entertainment, politics, sport, and tech. The data is in CSV format and can be used for a variety of natural language processing tasks, such as text classification, sentiment analysis, and topic modeling. Additionally, the dataset is small, which makes it an excellent choice for training and testing models quickly. Each article in the collection includes a title, a brief synopsis, and the article's full content. The articles were published between 2004 and 2005 and are identified by the category to which they belong. The dataset can be used for text categorization, sentiment analysis, and topic modeling, among other natural language processing applications.

**2.6.2. Ask a manager 2021 Salary Survey Dataset**

Ask A Manager is an advice website for workplace-related questions. Earlier in 2021, the blog site published a Manager Salary Survey where respondents anonymously shared their current salary among other biographical details. The survey and the resultant data are interesting in several ways. One reason is that the responses are anonymous, hence there is a low risk of exposing participants to personal information. Also, the dataset is large (N=26538) and publicly available.

## Storage Medium

For this project we will are considering using the amazon s3 bucket. We will store the data files in S3 bucket which will act as a initial storage for our data. We will also be using snow pipe to integrate snowflake which will be useful in storing and fetching data from S3 bucket to the snowflake database, and S3 bucket also provides us with the functionality where we can use S3 events to trigger actions based on changes to the data files. Enormous amounts of data can be handled with ease by Amazon S3, which is very scalable. This makes it perfect for tasks involving the management and storage of massive amounts of data because Amazon S3 is built for 99.99% durability, our data will always be accessible whenever we need it and will be highly resilient to failures. For projects that demand high availability and reliability, this is essential. A cost-effective storage option is Amazon S3, which charges according to the volume of data kept and the level of access needed. It is therefore a cost-effective choice for undertakings with tight budgets.

Amazon S3 integrates seamlessly with other AWS services, such as Amazon Elasticsearch, Amazon Athena etc. making it easy to build a complete data storage and search solution.

## Storage Security

Amazon S3 provides a variety of security features and controls to help you protect your data stored in the service.

We have access to several tools through Amazon S3, including Identity and Access Management (IAM) rules, Bucket policies, Access Control Lists (ACLs), and pre-signed URLs (Uniform Resource Locator), to manage who has access to our data. We may also impose fine-grained access controls on individual items or buckets using these protocols, as well as give or revoke permissions to individuals, groups, or roles.

We have a lot of options with Amazon S3 to manage network-level access to our data. We can use Amazon S3 Block Public Access to ensure that S3 buckets or objects are not publicly accessible, create Access Points to enable network regulations and routing rules, or use Virtual Private Cloud (VPC) to isolate our S3 resources.

## Storage Costs

Storing data in your S3 buckets costs money. The price you pay is determined by the size, duration, and storage class of your items as well as how often you access those items during the month. To track access patterns and automate the movement of objects between access tiers, you must pay a monthly monitoring and automation fee per item stored in the S3 Intelligent-Tiering storage class. There are no retrieval fees in S3 Intelligent-Tiering, and there are also no additional tiering fees when items are moved across access levels.

For this project, the S3 standard package would suffice. It costs $0.023 per GB per month, for the first 50TB of data. The cost increases as the data storage capacity increases but that is irrelevant since this project doesn’t use more than 50TB of data.

# Algorithms & Analysis / ML Model Exploration & Selection

## Solution Approach

**INSTRUCTIONS**

Provide a detailed discussion of the solution approach. Include discussions on any of the following:

1. Systems Architecture
2. Systems Security
3. Systems Data Flows
4. Algorithms & Analysis
5. Machine Learning (delete this subsection for non-machine learning projects.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

### Systems Architecture

//Planned an architecture which was perfect suitable for future tasks on this project...Let’s elaborate more on that.

### Systems Security

//Ravinder has idea on this Security work

### Systems Data Flows

### Algorithms & Analysis

## Machine Learning

**INSTRUCTIONS**

For Machine Learning projects discuss the model exploration and selection process. Delete this report subsection for non-machine learning projects.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

### Model Exploration

### Model Selection

# Visualizations / ML Model Training, Evaluation, & Validation

## Overview

**INSTRUCTIONS**

Provide an overview of what was accomplished during Sprint 4. Focus visualizations for non-machine learning projects.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

## Visualizations

## Machine Learning

**INSTRUCTIONS**

For Machine Learning projects, discuss your approach to the following with respect to the ML Model:

1. Training,
2. Evaluation, and
3. Validation of the ML Model.

Delete this report subsection for non-machine learning projects.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

### Model Training

### Model Evaluation

### Model Validation

# Findings

**INSTRUCTIONS**

Discuss the major findings of the project.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

# Summary

**INSTRUCTIONS**

Summarize the overall project and results for the reader. What did you discover, prove, disprove, etc.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

# Future Work

**INSTRUCTIONS**

This is critical section of the report. Propose future follow-on work or next step(s) for the project.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

Appendix

Appendix A: Glossary

|  |  |
| --- | --- |
| Term | Definition |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**INSTRUCTIONS**

Place all terms which require definitions in the Appendix A: Glossary.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

Appendix B: GitHub Repository

Overview

**INSTRUCTIONS**

Provide a GitHub Link and the README.MD content. Do not just provide a link to the GitHub repository but provide a narrative paragraph which introduces the project. This section should mirror the look and feel of a well-documented professional GitHub site.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

GitHub Repository Link

GitHub Repository Contents

Appendix C: Risks

Sprint 1 Risks

**INSTRUCTIONS**

Include the risk table associated with the Sprint. Below the risk table provide a narrative description of how the risks and mitigation plans were identified, what the team got correct, what the team could have done differently, how accurate was the team in identifying the risks, did the team encounter any unanticipated risks, etc. Think of this writeup as a “lessons learned” that you would like to pass along to any project team thinking of doing a similar project.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

Sprint 2 Risks

**INSTRUCTIONS**

Include the risk table associated with the Sprint. Below the risk table provide a narrative description of how the risks and mitigation plans were identified, what the team got correct, what the team could have done differently, how accurate was the team in identifying the risks, did the team encounter any unanticipated risks, etc. Think of this writeup as a “lessons learned” that you would like to pass along to any project team thinking of doing a similar project.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

Sprint 3 Risks

**INSTRUCTIONS**

Include the risk table associated with the Sprint. Below the risk table provide a narrative description of how the risks and mitigation plans were identified, what the team got correct, what the team could have done differently, how accurate was the team in identifying the risks, did the team encounter any unanticipated risks, etc. Think of this writeup as a “lessons learned” that you would like to pass along to any project team thinking of doing a similar project.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

Sprint 4 Risks

**INSTRUCTIONS**

Include the risk table associated with the Sprint. Below the risk table provide a narrative description of how the risks and mitigation plans were identified, what the team got correct, what the team could have done differently, how accurate was the team in identifying the risks, did the team encounter any unanticipated risks, etc. Think of this writeup as a “lessons learned” that you would like to pass along to any project team thinking of doing a similar project.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

Sprint 5 Risks

**INSTRUCTIONS**

Include the risk table associated with the Sprint. Below the risk table provide a narrative description of how the risks and mitigation plans were identified, what the team got correct, what the team could have done differently, how accurate was the team in identifying the risks, did the team encounter any unanticipated risks, etc. Think of this writeup as a “lessons learned” that you would like to pass along to any project team thinking of doing a similar project.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

Appendix D: Agile Development

Scrum Methodology

**INSTRUCTIONS**

Provide a narrative of the team efforts in using a scrum methodology for a data analytics engineering project. Describe how easy/difficult was it to adapt to the Scrum methodology. Did the team conduct a daily scrum? If not, how often did the team conduct a scrum. Describe how easy/difficult it was to use the YouTrack tool to manage the project. Don’t be limited to just these questions. Think of this writeup as a “lessons learned” that you would like to pass along to any project team thinking of doing a similar project.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

Sprint 1 Analysis

**INSTRUCTIONS**

Provide a narrative of the team’s efforts during this Sprint. Be sure to include – but not be limited to – how the team identified the User Stories, how well the team performed with the various tasks, how easy/difficult it was for the team to manage their activities during the Sprint, what did the team do correct, what could/should the team have done differently, etc. Think of this writeup as a “lessons learned” that you would like to pass along to any project team thinking of doing a similar project.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

Sprint 2 Analyis

**INSTRUCTIONS**

Provide a narrative of the team’s efforts during this Sprint. Be sure to include – but not be limited to – how the team identified the User Stories, how well the team performed with the various tasks, how easy/difficult it was for the team to manage their activities during the Sprint, what did the team do correct, what could/should the team have done differently, etc. Think of this writeup as a “lessons learned” that you would like to pass along to any project team thinking of doing a similar project.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

Sprint 3 Analysis

**INSTRUCTIONS**

Provide a narrative of the team’s efforts during this Sprint. Be sure to include – but not be limited to – how the team identified the User Stories, how well the team performed with the various tasks, how easy/difficult it was for the team to manage their activities during the Sprint, what did the team do correct, what could/should the team have done differently, etc. Think of this writeup as a “lessons learned” that you would like to pass along to any project team thinking of doing a similar project.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

Sprint 4 Analysis

**INSTRUCTIONS**

Provide a narrative of the team’s efforts during this Sprint. Be sure to include – but not be limited to – how the team identified the User Stories, how well the team performed with the various tasks, how easy/difficult it was for the team to manage their activities during the Sprint, what did the team do correct, what could/should the team have done differently, etc. Think of this writeup as a “lessons learned” that you would like to pass along to any project team thinking of doing a similar project.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

Sprint 5 Analysis

**INSTRUCTIONS**

Provide a narrative of the team’s efforts during this Sprint. Be sure to include – but not be limited to – how the team identified the User Stories, how well the team performed with the various tasks, how easy/difficult it was for the team to manage their activities during the Sprint, what did the team do correct, what could/should the team have done differently, etc. Think of this writeup as a “lessons learned” that you would like to pass along to any project team thinking of doing a similar project.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

Reference

### **1  Zaidi, E., et al.: Data catalogs are the new black in data management and analytics (2017).** [**https://www.gartner.com/en/documents/3837968/data-catalogs-are-thenew-black-in-data-management-and-a**](https://www.gartner.com/en/documents/3837968/data-catalogs-are-thenew-black-in-data-management-and-a)

2  Dibowski, H., et al.: Using semantic technologies to manage a data lake: data catalog, provenance and access control, p. 17 (2020)

3 Labadie, C., et al.: Fair enough? Enhancing the usage of enterprise data with data catalogs. In: 2020 IEEE 22nd Conference on Business Informatics (CBI), vol. 1, pp. 201–210, June 2020.

4 M. D. Wilkinson et al., “The FAIR Guiding Principles for scientific data management and stewardship,” Sci. Data, vol. 3, p. 160018, Mar. 2016, doi: 10.1038/sdata.2016.18.

5. <https://www.oracle.com/big-data/data-catalog/what-is-a-data-catalog/>.

6. Wells, D. (2019). Introduction to Data Catalogs.

7. Hendler, J. (2014, December 1). Data integration for heterogenous datasets. Big data.

Retrieved February 12, 2023, from <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4276119/>

8. Marisa. (2022, July 5). OpenMetadata 0.11 release. Medium. Retrieved February 12, 2023, from <https://blog.open-metadata.org/openmetadata-0-11-release-8b82c85636a>

9. <https://www.elastic.co/what-is/elasticsearch>

10. **Antti Ala-Ilkka** | October 14, 2019 | from: <https://www.addsearch.com/blog/faceted-search/>

11. Moutjo Sen |Aug 30,2018 | From : <https://medium.com/@mourjo_sen/a-detailed-comparison-between-autocompletion-strategies-in-elasticsearch-66cb9e9c62c4>

12. Haslhofer, B., & Klas, W. (2010). A survey of techniques for achieving metadata interoperability. *ACM Computing Surveys (CSUR)*, *42*(2), 1-37.

13 A. Kumar, A. Bandyopadhyay, H. Bhoomika, I. Singhania, and K. Shah, “Analysis of network traffic and security through log aggregation,” International Journal of Computer Science and Information Security (IJCSIS), vol. 16, no. 6, 2018.

14.  R. Kuc and M. Rogozinski, ´ Elasticsearch Server. Packt Publishing Ltd, 2016.

15. C. Gormley and Z. Tong, Elasticsearch: The Definitive Guide: A Distributed Real-Time Search and Analytics Engine. ” O’Reilly Media, Inc.”, 2015.

16. P. McDermott, ‘‘Building open government,’ Government Inf. Quart., vol. 27, no. 4, pp. 401–413, Oct. 2010.

17. W3C. (2020). Data Catalog Vocabulary (DCAT)–Version 2. W3C

**This page intentionally left blank**