**Data Science - Design Specification**

**V2.5.7.1**

|  |  |
| --- | --- |
| **Author** | Bhagyaram Subramani |
| **Created On** | 12-June-2024 |
| **Reviewed By** | Dr. Srinivasa Vegi |
| **Reviewed On** |  |

**Change Record**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version No** | **Date** | **Sections Affected** | **Reason/Remarks** | **Author** |
| v1.0 | 22-Feb-2021 |  | Initial version | Bhagyaram S |
| v2.0 | 29-Sep-2021 |  | Update | Bhagyaram S |
| v2.1 | 31-Oct-2022 |  | Update | Bhagyaram S |
| v2.5 | 16-Nov-2022 |  | Update | Bhagyaram S |
| v2.5.5 | 20-Oct-2023 |  | Sep 2023 Release | Bhagyaram S |
| v2.5.7 | 05-Apr-2023 |  | Mar 2024 Release | Bhagyaram S |
| V2.5.7.1 | 12-Jun-2023 |  | June 2024 Release | Bhagyaram S |

# Table of Contents

[Table of Contents 2](#_Toc273975597)

[1. Description 3](#_Toc179455066)

[1.1. Purpose 3](#_Toc889536336)

[1.2. Scope 3](#_Toc378899221)

[2. Work flow 5](#_Toc47052923)

[2.1. Landing Area to Level 1 processing - Unstructured Data Extraction 5](#_Toc657310748)

[2.2. Level 1 Processing - Product Segmentation 5](#_Toc1595179486)

[2.3. Level 2 Processing - PSLU derivation 5](#_Toc1088035432)

[2.4. Semantic Search Engine 5](#_Toc1548678933)

[3. Landing Area to L1 Area Processing - Unstructured Data Extraction 5](#_Toc736709235)

[3.1. Fetch data from source address 6](#_Toc1066744568)

[3.2. Extract raw content from the documents 6](#_Toc285819150)

[3.3. Segregate data 6](#_Toc1408899304)

[3.4. Read Reference file 6](#_Toc116225057)

[3.5. Document classification 6](#_Toc1641238533)

[3.6. Extract Header information 7](#_Toc1742091012)

[3.7. Extract detail tabular information 7](#_Toc1150871710)

[3.8. Header Vs Detail Data collation 7](#_Toc1748825152)

[3.9. Data profiling, Quality Check& Standardization 7](#_Toc832395069)

[3.10. Transformation & Output generation 7](#_Toc180862925)

[3.11. Data Movement 8](#_Toc1315194215)

[4. L1 Area Processing - Product Segmentation 8](#_Toc937041711)

[4.1. Derive State Codes 8](#_Toc56785569)

[4.2. De-Dupe Vendor Names 8](#_Toc901408554)

[4.3. Base Product Derivation 8](#_Toc1278569971)

[4.4. Product Segmentation 8](#_Toc437409441)

[4.5. UNSPSC Mapping 9](#_Toc1013887793)

[4.6. PSLU / hunterAI ID Derivation 9](#_Toc2137700744)

[5. L2 Area Processing – Pricing derivation 10](#_Toc83602487)

[5.1. Savings Calculation 10](#_Toc2147124973)

[6. Semantic Search Engine 10](#_Toc925085344)

[7. Sentiment Analysis 11](#_Toc510194734)

[8. Audit & Approvals 12](#_Toc819559300)

[9. Quality Monitoring 12](#_Toc11113989)

# Description

## Purpose

This document describes the software design workflow for the implementation of data extraction from Unstructured SOW, Invoice & PO and Structured invoices.

## Scope

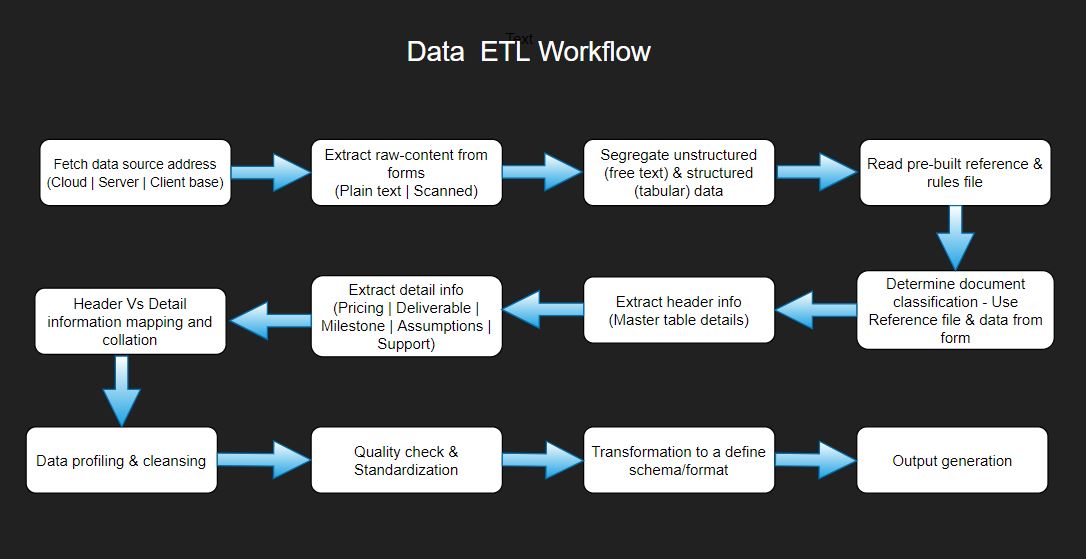
This describes the software design decisions, data extraction techniques & algorithms and the various processes in the Data Science workflow for the implementation of data extraction from Unstructured SOW, Invoice & PO and Structured invoices. The intended audience of this document is the developers, designers, and software testers of GPO Health team.

|  |  |  |
| --- | --- | --- |
| **Step** | **Process** | **Description** |
| **1** | **ETL – Extract, Transform & Load** | The overall process comprises of the below 6 stages for Unstructured data: 1. **Extraction** - Extract the data in unstructured raw format from the various source systems - forms uploaded into the cloud, secure server, directly connected to client base.  2. **Profiling** - The next step uses data profiling tools and techniques to make an initial assessment of the data to understand its overall quality challenges and anomalies. This step primarily uses patterns, trends, and algorithms to produce a picture of the overall data quality. 3. **Cleansing** a. Identify and remove/modify incomplete, incorrect, inaccurate, and irrelevant data. These processes involve removing typographical errors or validating and correcting values against a known list of entities. The validation may be strict (such as rejecting any amount value extracted containing only alphabets from scanned forms) or fuzzy (correcting records that partially match existing and known records). The primary intention is data consistency across the field attributes. b. Eliminate redundant/duplicate data to provide consistency and reduce storage space 4. **Standardization** - Execute a series in-built data quality checks and rules against the data (ex. total amount = sum of line amounts) to obtain the required degree of data consistency and feature generation. 5. **Transformation** - Transform the data into the required pre-defined format/schema as recognized by the repository.  6. **Load** – Pass on the transformed data for ETL load.  Structured data is directly consumed from the database. |
| **2** | **Categorization** | 1. **Vendor Deduping**    1. Standardization of vendor names 2. **Base Product Identification**    1. Text pre-processing    2. Feature Engineering for Label preparation    3. Gram derivation    4. Domain based gram association    5. Clustering & Classification 3. **Product Segmentation**    1. ML Models for item classification    2. Segment identification 4. **OEM Identification**    1. OEM Identification from Description & Vendor 5. **UNSPSC Mapping**    1. Association of segments to UNSPSC level 6. **Vendor Based Segmentation**    1. Broad level segmentation using vendor categories 7. **PSLU Derivation**    1. Pattern derivation using 6-18 attributes depending on Base Product    2. PSLU – Internal Product Code derivation 8. **Savings Calculation**    1. Pricing derivation grouping by PSLU    2. Potential Savings determination at PSLU level |
| **3** | **Semantic Search Engine** | Semantic Search application is a WSGI web application framework that takes input string (product description) from the user, runs the ML model in the background to throw out the predicted results. |
| **4** | **Audit & Approvals** | The "processed" data will be approved by the relevant authority to be locked down in the repository. Controls exist as to who can change locked down data. |
| **5** | **Quality Monitoring** | Data quality is as an ongoing process to track the of data quality over time to avoid data decay, which fundamentally occurs overtime due to changes not being captured or system upgrades. Typically, the ongoing process is to execute the data quality check module on a scheduled basis (ideally after inception of a new vendor/bi-weekly/monthly) to auto-correct the variations based on pre-defined business rules. |

# Work flow

## Landing Area to Level 1 processing - Unstructured Data Extraction

1. Unstructured data is executed through the Data ETL workflow – Data Extraction, profiling, transformation and output generation
2. Transformed data is passed on to ETL for data push into L1.



## Level 1 Processing - Product Segmentation

Data is picked up from Level 1 by customer, post Talend ETL for the following operations.

1. Derive State Codes
2. Normalize vendor Names
3. Base product derivation
4. Product Segmentation
5. UNSPSC Mapping
6. Customer & Vendor Segmentation
7. PSLU & Internal product description derivation

## Level 2 Processing - PSLU derivation

Data is fetched from Level 2 post Talend ETL for the following operations

1. Savings Calculation at Detail and Summary Level

## Semantic Search Engine

Semantic Search application is a WSGI web application framework that takes input string (product description) from the user, runs the ML model in the background to throw out the predicted results. Primary intention is to help the user decide on the least/optimum “Unit Purchase Price” of relevant products along with PSLU information with the minimum and maximum price range.

# Landing Area to L1 Area Processing - Unstructured Data Extraction

## Fetch data from source address

* Connectivity to the input data source is established – cloud data storage, secure server data hosting warehouse or client data base.
* Location addresses of all the documents to be extracted are fetched.

## Extract raw content from the documents

* Data extraction framework built for PDF. Other document types in MS Word are converted systematically into PDF form.
* The type of document is identified – Searchable free text or Non searchable scanned document
* Below open-source python packages are used for raw data extraction depending on the document type
* Searchable document - PyMuPDF&PDFPlumber
* Non searchable - Google tesseract & AWS Textract

## Segregate data

* Extracted data is separated into Structured (tabular data format – rows & columns) & Unstructured free text.
* For searchable documents, PyMuPDFis used to extract full text &PDFPlumberto identify structured data.
* For scanned documents,
  + Vertical and horizontal lines are identified using OpenCV package. System provided intelligence to identify structural data by analysing these lines and data capsuled between them.
  + For forms without vertical and horizontal lines, AWS Textract is used to extract structural content

## Read Reference file

* Pre-built reference file is read to absorb the rulesets for document classification to identify type of document – SOW/Invoice/PO
* Reference file is hosted as MS Excel workbook
* Defined “Schema Mapping” is read to identify the various sub-table type from all the tabular/structural content and map the fields one-to-one into the desired repository recognizable format. Sub table types include – Pricing, Milestone, Assumption, Deliverable & Support details.

## Document classification

* Free text extracted as described in section 3.3 is flown through the below pre-processed algorithms
  + Special & Unicode charactersremoved
  + Numerical information removed
  + Words of highest significance from the entire document are filtered using Word-to-vector &TFIDF
* Filtered words are compared for similarity from the rules/words in the reference file to identify the document class

*This would be later moved into a Machine Learning space once enough sample data is available*

## Extract Header information

* Data is extracted field by field using the below 3 techniques. Numpy, pandas and regular expression algorithms are used for extraction.
  + By co-ordinates (exact location based on x-y coordinates in a scanned form)
  + Defined ruleset based on neighbouring fields (from all four directions – left-to-right & top-to-bottom)
  + Field patterns & type of information
* Scope summary is extracted using page rank algorithm and cosine similarity against the various sentences from the scope column

## Extract detail tabular information

* 5 different sub-table types where the detail information is stored are mentioned in section 3.4
* Each tabular or structural content retrieved from the document is pre-processed into python DataFrameformat
* In case of flowing tables across pages, data structure from previous page table is analysed to fetch column headers on structural match
* Data across blank headers are validated and merged with existing headers based on data consistency
* Table headers or columns is compared to the pre-defined “Schema mapping” ruleset as described in section 3.4 to identify the sub-table type

## Header Vs Detail Data collation

* Below header level information are passed onto the detail information for linkage
  + Filename
  + Customer Name
  + Supplier Name
  + SOW #
* For handling files with multiple invoices (from different SOW numbers) stored in the same document, the relational order of data & data consistency check is established to map the header information to the corresponding detail extract

## Data profiling, Quality Check& Standardization

Below are the data profiling, quality check& standardization functions used to attain data consistency. Identify and remove/modify incomplete, incorrect, inaccurate, and irrelevant data. Execute a series in-built data quality checks and rules against the data (ex. total amount = sum of line amounts) to obtain the required degree of data consistency and feature generation.

## Transformation & Output generation

* Reference file mentioned in section 3.4 hosts the “Schema Mapping” for both header and detail level information as per the repository recognizable format
* Data extracted (header & detail) as mentioned in sections 3.6, 3.7 & 3.8 are mapped one-to-one against the schema to generate the final output.

## Data Movement

* Unstructured data is executed through the Data ETL workflow – Data Extraction, profiling, transformation and output generation
* Structured data is segregated into different folders based on Customer
* Output from Unstructured format and Structured format are pushed to L1 Area

# L1 Area Processing - Product Segmentation

Algorithms - Spacy, NLTK, Word2Vec/TFIDF, Scikit, sentence\_transformers, BERT, Pandas, Numpy, Tensorflow, Keras, Regular Expression

## Derive State Codes

* State Codes are derived based on the Hospital Name Vs Location mapping for all the records
* Hierarchy followed – Branch Office, Regional Office, Head Quarters

## De-Dupe Vendor Names

* Text Pre-processing & noise removal
* Text distance-based algorithm to cluster vendor names
* Standardize the cluster names
* Multi Name dimension common clustering

## Base Product Derivation

* Text pre-processing
* Noise removal (Stop words, punctuation, cleaning)
* Normalization/Pruning (Tokenization, Stem, Lemmatization)
* Standardization (RegEx, Lookup). For ex. 1st - first
* Feature Engineering for Label preparation
* n-gram derivation from product description by domain
* Commodity clustering using synonyms &grams

## Product Segmentation

* **Commodity/Level4 categorization**
  + CDW Categorization – CDW/SHI open source extraction based knowledge base used in keyword based & self source developed word embedding model to determine the categorization at CDW/SHI based hierarchy.
  + Deep Learning prediction model
    - Individual Deep Learning models developed for each Segment.
    - Transformers based deep learning model using self generated tokenizing vectors for IT product commodity identification
    - Medical Embeddings from PubMed and BioBERT used for categorization of Medical Equipment, Supplies, Pharmaceuticals, Laboratory Testing Equipment & Apparels.



* + Synonym prediction model - Synonym based product commodity identification Level 4 across all segments based on in house training data of ~750k synonyms
  + Ensemble Model - Ensemble model based on CDW Categorization, DL threshold scoring, synonym model for final commodity Level 4 identification
* **Segment/Level1 categorization**
  + NAICS categorization - Identification of segments Level 1 for all the items where scoring for Commodity derivation is below the pre-determined threshold levels, based on NAICS Level1 codes extracted from open source search tools like OpenAI
* **Software Categorization**
  + Module to categorize software grouping based on synonyms/keywords from in- house-curated software database from web scraping from open source databases like Onet database
* **OEM Identification**
  + Module to identify and extract the OEM by looking at the Item Description and Vendor

## UNSPSC Mapping

* UNSPSC 4 level categorization (Segment/Family/Class/Commodity) process using Text Distance Infographics - Token grams/Sequence based on Word Embeddings from the derived Level 1 & Level 4 from the earlier mentioned processes in Section 4.4.

## PSLU / hunterAI ID Derivation

* Pattern derivation using n attributes – by Commodity
* Feature extraction (Regular Expressions) from description
* NGram from Base Product
* NER - Named Entity Recognition using synonyms
* PSLU based on the features extracted from product description depending on the Base Product. Each Commodity has a specific number of attributes/features extracted.
* List of Commodities for hunterAI ID derivation
  + Laptop/Notebook Computers
  + Desktop Computers
  + Workstation Computers
  + Tablet Computers
  + Printer
  + Urological Stents
  + Urological Implants
  + Coronary Stents
  + Fresh Juice
  + Medical Screws
* PSLU/hunterAI ID derivation by
  + Product – GPO level
  + System/Member level
  + ABG – Customer based Business Grouping level

# L2 Area Processing – Pricing derivation

Algorithms - Spacy, NLTK, Pandas, Numpy, Regular Expression

## Savings Calculation

* Min-Median-Max Unit price is derived based on the products from each PSLU/hunterAI ID
* Min & Max from each PSLU/hunterAI ID category forms the basis points for Unit Price recommendation based on a time period approach at Commodity Level. Each commodity can be self coded with the required time period for Min & Max price derivation.
* Pricing numbers are updated at detail and summary level
* Savings are derived at the below levels
  + Product level
  + System/Member level
  + ABG – Business Grouping member level

# Semantic Search Engine

Algorithms - sentence\_transformers, faiss, flask, sentence\_transformers, faiss, flask, pandas, numpy, Spacy, Regular Expression

This is an UI to the end-user to input a product description to compare the Unit Price of similar products.

Approach

* User input converted into BERT embeddings
* Keyword extraction using a cluster of algorithms
* Pre-trained word embedding model with the existing product catalogue
* Ranking Matrix – Soft Vote for keywords
* Multiple Text Distance Infographics methods to find relevant products

Semantic Search advantages

* Searcher intent based on behaviour
* Query context based on historical data
* Word-to-word relationship
* Faster and accurate

# Sentiment Analysis

Algorithms - Pandas, Spacy, Regular Expression, SciKit Learn, Openai

Employee and Customer Reviews were analysed for sentiment extraction. From each review, depending on the type, pre-defined nature of text was established based on overall and entity-based parameters. Part texts were extracted as applicable for every nature of the message. Sentiment related words and sentiment type was extracted based on various parts of the message based on the nature of text.

Approach

* Pre Processing - Tokenization, Normalization & Punctuation removal
* Contraction Handling
* Context preservation
* POS tagging
* Feature extraction using BERT
* Aspect-Based Sentiment Words Extraction using pre-trained LLM models
* Words based Sentiment detection using pre-trained LLM models

Sentiment Analysis advantages

* **Understand Customer Sentiment:** Analyze reviews to gain insights into how customers feel about your product, service, or brand. This helps you identify areas of strength and areas for improvement.
* **Improve Product Development:** Identify trends in positive and negative sentiment to guide product development efforts. Focus on features and functionalities customers appreciate and address common pain points.
* **Enhance Customer Service:** Sentiment analysis can help you prioritize customer service tickets based on urgency and emotional tone. Identify frustrated customers and proactively reach out to address their concerns.
* **Monitor Brand Reputation:** Track sentiment over time to understand how your brand perception is evolving. This allows you to identify potential reputation issues and take corrective actions.
* **Gain Competitive Intelligence:** Analyze customer reviews of your competitors to understand their strengths and weaknesses. Use this information to develop a competitive advantage for your own products or services.

# Audit & Approvals

The "processed" data will be approved by the relevant authority to be locked down in the repository. Controls exist as to who can change/updatethe locked down data.

# Quality Monitoring

Data quality is as an ongoing process to track the of data quality over time to avoid data decay, which fundamentally occurs overtime due to changes not being captured or system upgrades. Typically, the ongoing process is to execute the data quality check module on a scheduled basis (ideally after inception of a new vendor/bi-weekly/monthly) to auto-correct the variations based on pre-defined business rules.