Contents:

I. Market Intelligence:

Business Requirement:

A government buyer typically selects products based on the best offer price available on the GeM platform for a given set of specifications. However, this process does not necessarily ensure the best market price for the product, as the prices listed on GeM are not validated against broader market trends or performance.

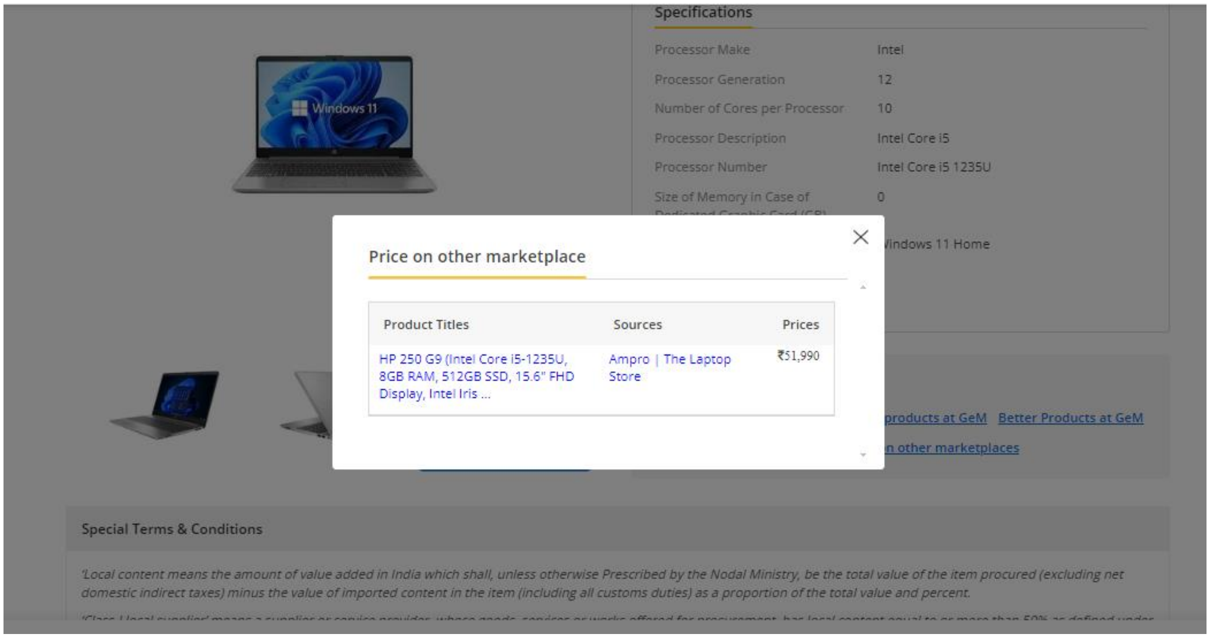
To address this, market intelligence data for specific categories should be obtained and made available to buyers. This would enable them to compare GeM-listed prices with market rates and leverage the GeM platform more effectively to achieve price parity. Additionally, this data would assist in cases where GeM prices exceed market rates, helping buyers discover the most competitive prices.

Proposed solution:

To enhance market intelligence, data from various e-marketplaces can be scraped using open-source Python libraries such as BeautifulSoup and Scrapy. These tools allow for efficient extraction of product listings, prices, and other relevant details from online platforms. By regularly scraping and updating market data, buyers can be empowered to make informed decisions by comparing GeM prices with real-time market trends. This automated approach will help identify price discrepancies and ensure better pricing accuracy within GeM.

Solution Consumption:

Currently, this solution is integrated into the Product Display Page (PDP) of the GeM portal, providing buyers with actionable recommendations directly where they view product detail.



Following are the stakeholders and processes where this solution can have impact:

- Buyer: The buyers will be able to access the competitive market values of similar products in other e-Marketplaces. This will help them to identify any product with better value in marketplace and whether the discovered prices are competitive or not in bids.

- Sellers: The sellers get the intelligence to mark the price of their product to be reflective of the market (indirect)

- GEM SPV: Will create competition among sellers and hence make the platform competitive among other e- Marketplaces with comparable listed prices.

Data:

To build this solution, required data consists of:

- Input: Product Id and Category Id
- Data is extracted from Google shopping, Google search, Amazon, Flipkart, Bing Search using scrapers (Python code).

Model Methodology:

This methodology outlines the complete process of optimizing the search for relevant products across e-marketplaces using specific keywords, scraping data, and preprocessing it to extract meaningful insights for buyers and sellers on the GeM platform. The aim is to help buyers find competitive prices and identify non-conforming products.

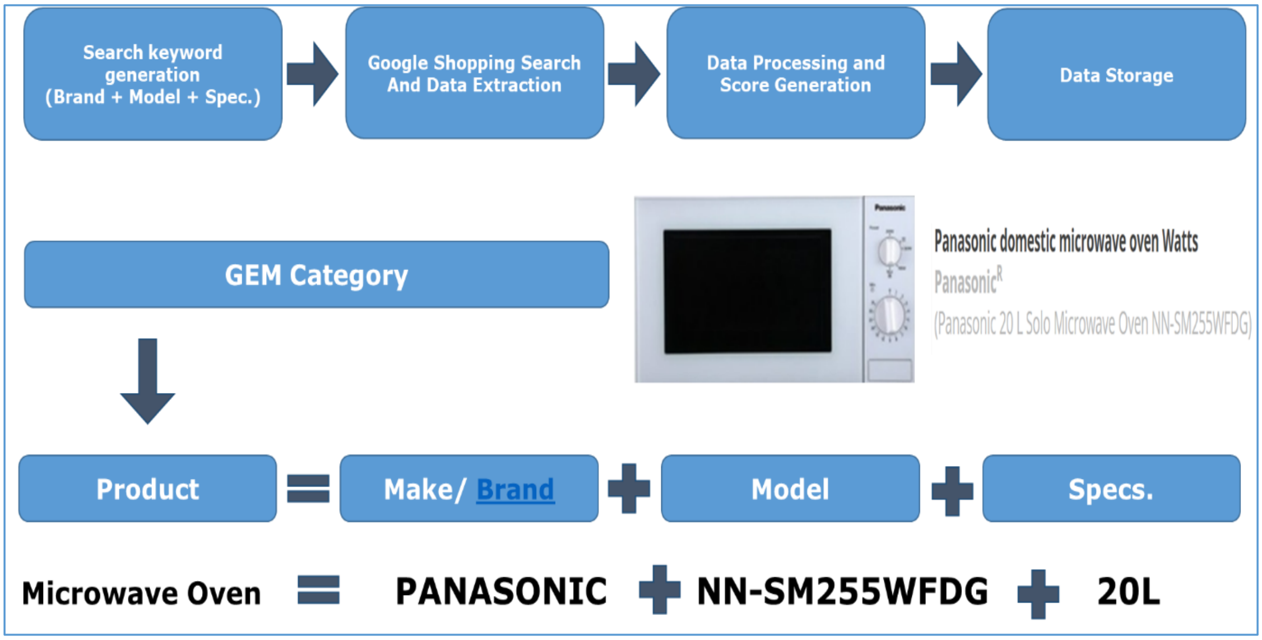
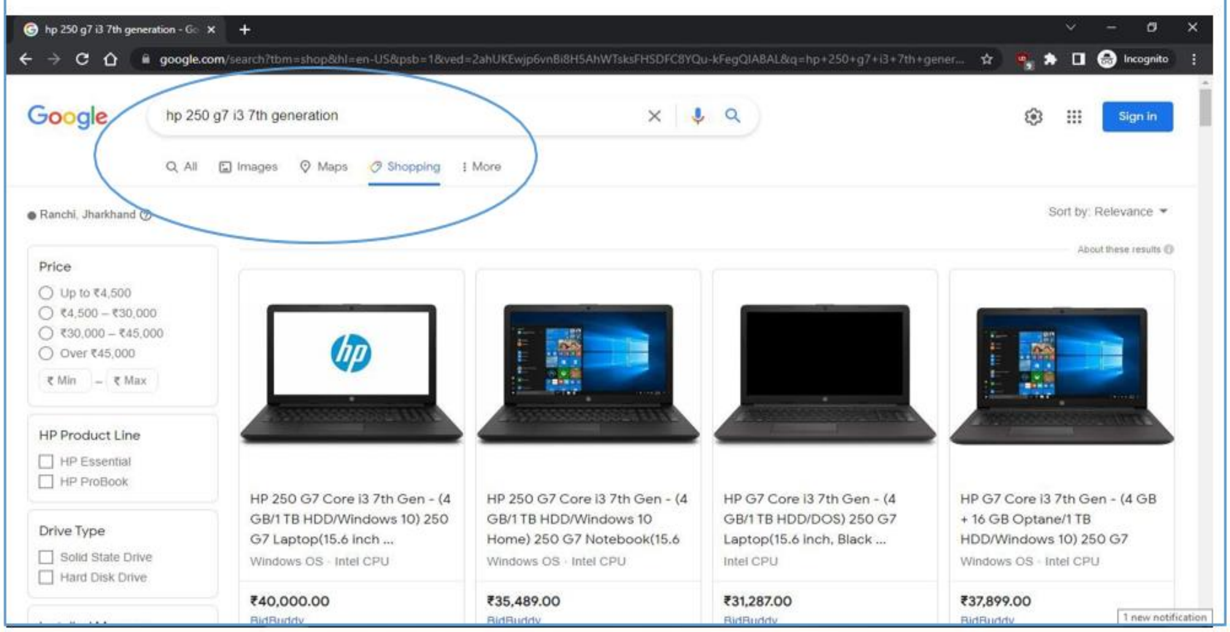


Image shows creating a search keyword. Example is for Microwave oven category



Steps In Model:

1. Creating Optimized Keywords for Product Search:
   - Goal: To create a precise and effective search string for product discovery on e-marketplaces like Amazon, Flipkart, and others.
   - Approach: Based on GeM data, important attributes such as make, model, and critical product specifications are extracted.

- Keyword Generation: After multiple rounds of experimentation, it was determined that a combination of the product's make, model, and a few key specifications (such as capacity, features, or technical specifications) forms the most optimized keyword for product discovery on search engines and e-marketplaces.
- Example: For a product like an air purifier, the keyword might look like: "Dyson air purifier model X, 300 sq ft, HEPA filter".

2. Using the Keyword for Search on E-Marketplaces:
    - Once the optimized search string is created, it is used to query various e-marketplaces (such as Flipkart, Amazon, or Snapdeal) via web scraping tools. The Python libraries used for this process include:
    - BeautifulSoup: For HTML parsing and extracting information from web pages.
    - Scrapy: For large-scale data extraction from multiple pages in a structured manner.

3. Scraping the Top 15 Relevant Products:
    - Goal: To limit the scope of scraping to the top 15 relevant products for each keyword search on e-marketplaces.
    - Reasoning: Based on extensive experimentation, the first 15 products on search results generally offer a good balance between relevance and diversity in pricing and features, which is critical for analysis.
    - Process: The solution scrapes product information like product title, price, brand, model, and specifications for the top 15 results, ensuring relevant comparison data from e-marketplaces.

4. Pre-processing the Scraped Data:
    - Goal: To clean the raw scraped data and ensure it is ready for meaningful analysis.
    - Removing Noise: Some scraped data may contain irrelevant or misclassified products, duplicate listings, or incomplete information. Such data is filtered out through pre-processing steps.
    - Standardizing Product Information: Product data from different sources may have slight variations in format or terminology. These are standardized to ensure comparability.
    - Handling Missing Data: For any missing fields such as price or brand, imputation techniques are applied where possible, or the incomplete entries are discarded.

5. Brand/Make Extraction:
    - Goal: To identify branded products within a category and focus on scraping data only for those products. Unbranded products are often not available or have inconsistent listings across e-marketplaces, making them unsuitable for comparison.

6. Keyword Formation:
    - The identified brands are used in the search string to ensure that relevant and branded products are scraped.
    - Example: After extracting the brand Dyson for an air purifier product on GeM, the search string could be "Dyson air purifier model X" to find matching products on e-marketplaces.

7. Handling Non-Branded Products:
   - Goal: To focus on branded products where comparable data exists across multiple e-marketplaces. Products that are unbranded often do not have consistent listings and may lead to gaps in data.
   - Process: During scraping, the solution filters out non-branded or generic products, as their market data cannot be reliably compared across platforms.

8. Using NLP (Fuzzy Wuzzy) to Find Similar Products:
   - Objective: Use Fuzzy Wuzzy, a powerful Natural Language Processing (NLP) technique, to identify and match similar products across different e-marketplaces, even when product names or descriptions differ slightly.
   - Fuzzy String Matching: When the scraped product names or descriptions from different e-marketplaces vary slightly due to inconsistencies in naming conventions or typographical differences, Fuzzy Wuzzy is used to find similar products based on their text similarity scores.
   - Implementation: The algorithm assigns a similarity score (from 0 to 100) based on the overlap of strings, allowing the system to compare and match products even when the names are not exactly the same.
   - Example: If GeM lists a product as "Dyson Air Purifier Model X", and an e-marketplace lists it as "Dyson Air Purifier - X Series", Fuzzy Wuzzy will calculate a high similarity score and match these products, enabling accurate price comparison.
   - Benefits: This technique helps improve product matching accuracy, especially in cases where product titles are slightly inconsistent across platforms. It ensures that all relevant products are included in the analysis, providing a more comprehensive view of market competitiveness.

9. Data Consolidation and Price Analysis:
   - Once the clean and standardized data is collected, it is consolidated into a structured format (for further analysis.
   - Price Comparison: The consolidated data is analyzed to identify price differences between GeM-listed products and their counterparts on e-marketplaces. This helps to spot instances where the GeM price is higher than the market, allowing buyers to make informed decisions.

Business Value:

1. Control High MRP & High Offer Prices Defined by Sellers: The solution directly addresses the issue of inflated Maximum Retail Prices (MRP) and high offer prices set by sellers on the GeM platform. By leveraging market intelligence data scraped from various e-marketplaces, the platform can compare sellers' listed prices with the broader market landscape. This will discourage sellers from setting unreasonably high MRPs and offer prices, knowing their pricing can now be benchmarked against market norms. Consequently, this promotes transparency and ensures fair pricing, which can drive more competitive and value-driven offers on the platform, benefiting both buyers and the platform itself.

2. Empower Buyers to Know the Right Price: The solution empowers government buyers with real-time access to market intelligence. With a clear comparison of product prices across multiple e-marketplaces, buyers can confidently assess whether the prices offered on GeM are competitive or inflated. This transparency in pricing helps buyers make more informed purchasing decisions, avoiding overpayment for products and optimizing procurement budgets. Empowered with data-driven insights, buyers can negotiate better deals and drive the selection of products that offer the best value for money.

3. Identify Non-Conforming Product Offerings Compared to the Market: By continuously monitoring market trends and prices from various e-marketplaces, the solution helps in identifying non-conforming or outlier product offerings on GeM. Sellers who offer products at prices significantly above market rates or who list products with mismatched specifications can be flagged. This identification process ensures that the platform maintains a high standard of product conformity and quality, reducing the risk of procurement inefficiencies caused by overpriced or subpar products. Ultimately, this will lead to a more trustworthy and competitive marketplace, ensuring that buyers are purchasing products that meet market standards in both price and quality.

II. Product Similarity:

Business Requirement:

In GeM portal, buyer can shortlist products based on a specific set of parameters and get the best price for the given set of parameters. However, no options / directional recommendations are provided to the buyers on possible parameter combination, that is not only an equivalent of the selected set of parameters but might also give a better value to cost proposition.

Presently, the GeM portal focuses solely on best price discovery rather than discovering the best value for the price. This means buyers are not informed about similar or better-featured products within the same price range as their selected product. As a result, buyers may not be making the most informed purchasing decisions and might be paying more than necessary.

To address this gap, GeM wants to develop a product similarity module to assist buyers. This module will offer a list of product suggestions with the same features and quality at a lower price or highlight better products available at the same price. By providing these recommendations, buyers can make more informed decisions, ensuring they receive the best value for their money.

With this solution, buyers will gain access to:

- **Similar Products at Lower Prices**: Identifying products with equivalent features at more competitive prices.
- **Better Products at Lower Price**: Highlighting options that offer superior features or quality without exceeding the buyer's budget.

By implementing the product similarity module, GeM aims to enhance the purchasing experience, ensuring buyers achieve optimal value for their expenditure.

Proposed solution:

The Product Similarity module is an intelligent, machine learning-powered system designed to identify products that are similar to a buyer's current selection. This module will recommend products either with a better price or superior value.

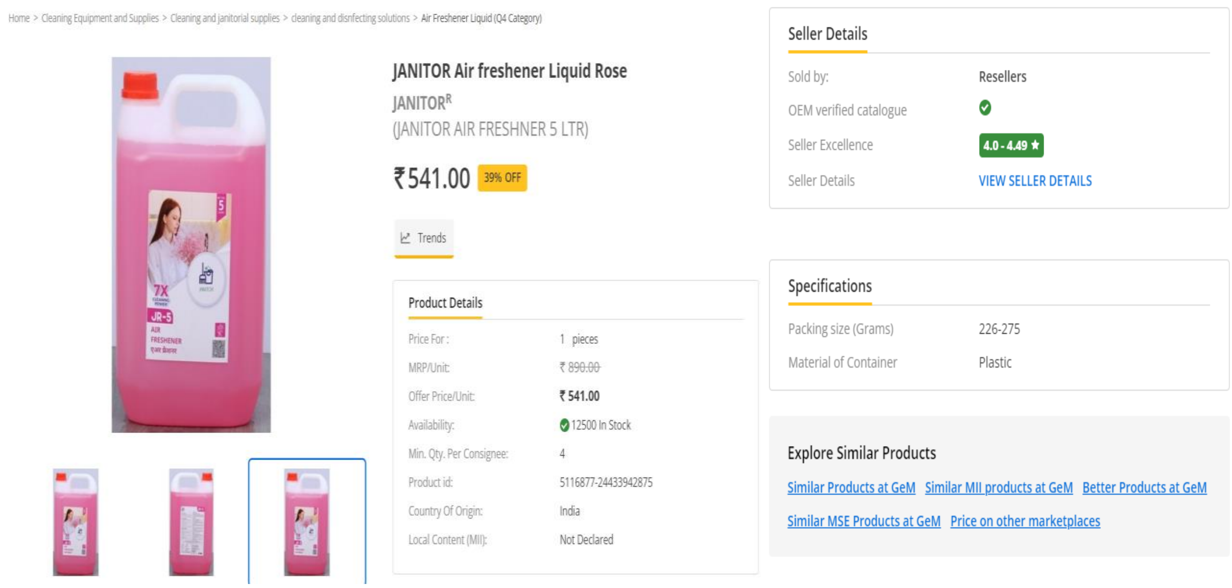
The objective of the product similarity module is to identify similar products for a given product of interested based on two primary criteria:

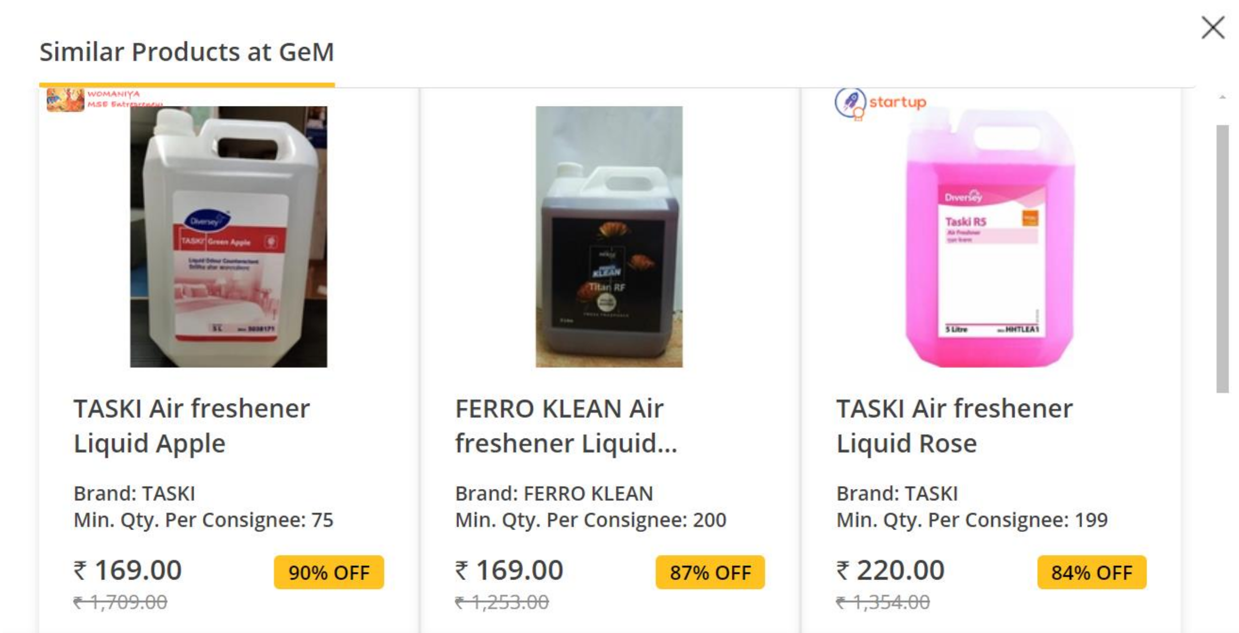
- The features of the shortlisted products are similar to the product of interest.
- The price of the shortlisted products is either lower than the product of interest or offers better value with upgraded components.

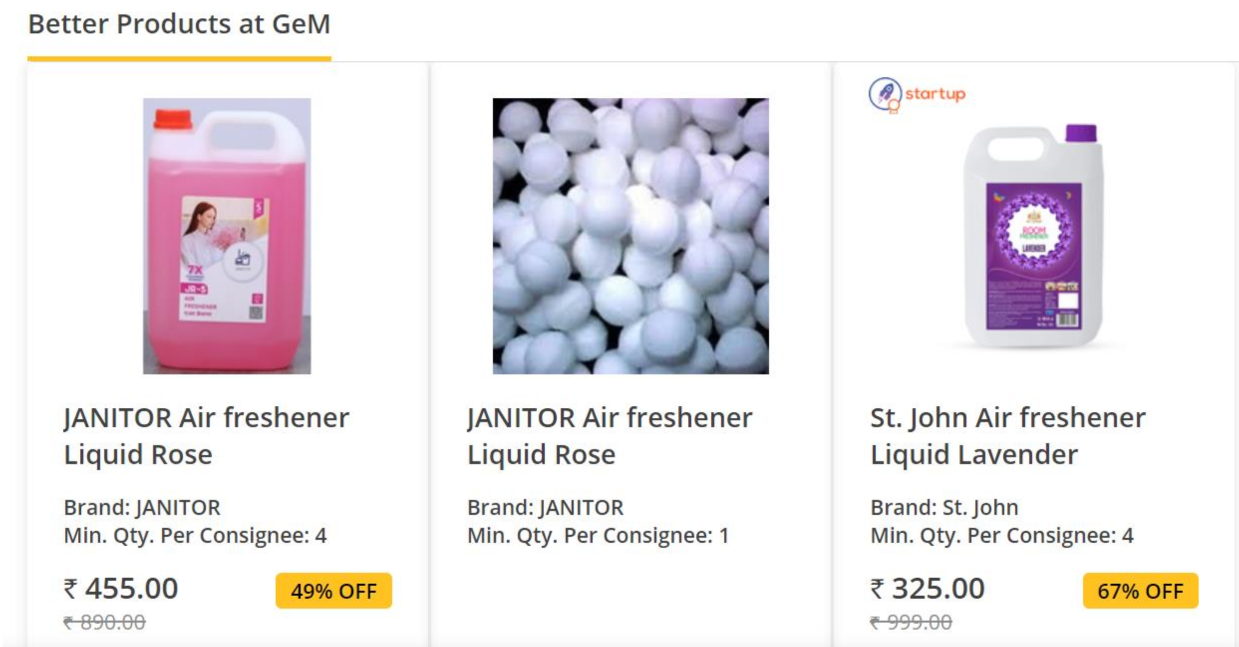
The Product Similarity module will transform the GeM portal from a platform focused solely on price discovery to one that also emphasizes value discovery. By offering intelligent, machine-learning-driven product comparisons, buyers will have access to a wider range of options that meet their needs both in terms of features and cost, ultimately enhancing their overall purchasing experience on the GeM portal.

Solution Consumption:

Currently, this solution is integrated into the Product Display Page (PDP) of the GeM portal, providing buyers with actionable recommendations directly where they view product detail.







Data:

To build this solution, required data consist of all the product id's in a category, along with its all-golden parameters, attributes within parameters and their respective order and offer price. The required data can be fetched from below tables and the list of golden parameters can be fetched from GeM API.

- Table: inb_variants
- Table: inb_browse_nodes
- Table: inb_stock
- Table: inb_catalog_attrs
- Table: ffm_orderitems
- API

Model Methodology:

To enable computation of feature similarity and importance, the product catalogue data - GeM data - will be one-hot encoded (better representation) and then we perform AI/ML model. This will help us understand the influence of each feature on price and also which attributes of a feature are similar. This analysis lays the foundation to understanding product value and product similarity.

The order of attribute importance is also considered in the analysis. Using data from the GeM API, we extract the attributes within a parameter and their order of importance. For parameters without predefined attribute orders, we can determine their order through business insights, research, or AI/ML approaches. The determined order of importance of attributes will be validated with the respective Category Owner/Manager.

To identify the impact of each parameter on the product price, we use AI/ML models. Every product has two types of parameters: golden and non-golden. In analyzing the relationship between price and parameters, we consider the golden parameters. The order of importance of these parameters is also validated with the respective Category Owner/Manager. The solution allows for flexibility in changing the order of importance of attributes and parameters as per the requirements of the category owner.

The order of importance of attributes is then scaled between 0 and 1. Using the scaled values of parameters and their importance, we calculate the score of product IDs by summing all the parameter scores. This process generates a score table for each product category.

With the product IDs scored based on features and attributes, we can compare them with other products. We can filter the results based on the offer price of the selected product, showing the most similar product with a lower price than the selected product. Additionally, since the scores are directly proportional to the features of the product, higher scores indicate better products. Thus, we can suggest better products within the price range of the selected product.

See the picture below to know the flow of processes in this solution:

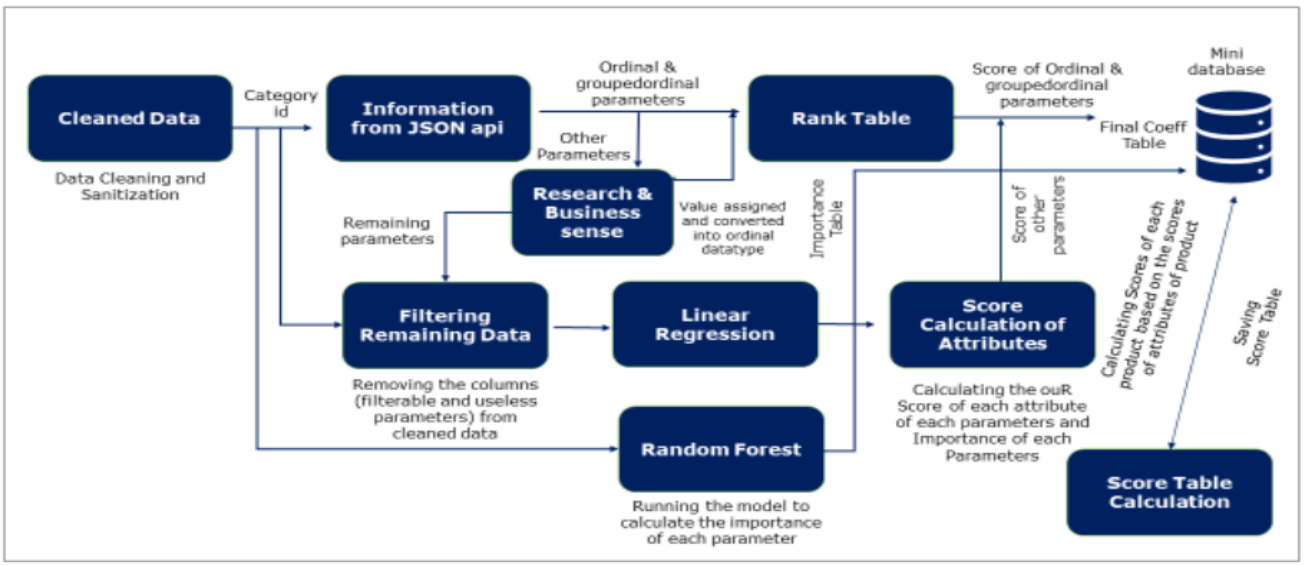


Figure 1: Solution Methodology

Steps In Model:

1. Extract the required data - All product/variant id's and associated its golden parameters of a category along with its offer price.
    - From 'stock' and 'Orderitem' table, we get the price of listed product and sold products respectively
    - For rest of the products we get the offer price from 'stock' table

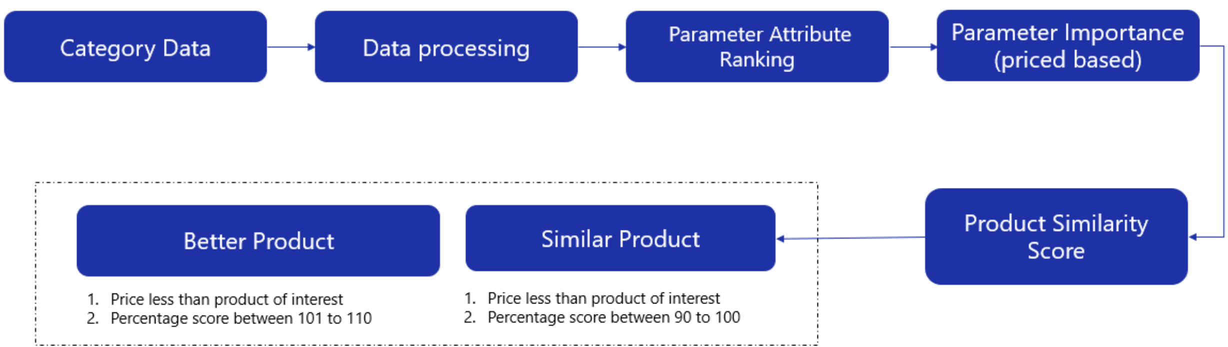
2. Clean data (pre-processing)
    - All parameters stored in YAML format, covert to python data-frame
    - Scrape the technical parameter names from API of GeM website and attributes within parameters along with their order/rank
    - Rename the parameters
    - Select only the golden parameters for further analysis
    - Missing value treatment

3. Attributes which don't have order provided, use business knowledge; if no business knowledge available, need to research and if both these ways don't work then we use AI/ML model to get order. Parameters which have data type 'group ordinal', 'ordinal', 'Boolean', and 'numeric and measurable' order has provided. Only for 'enumerable' data type don't have the order provided in API.
   - Categorize parameters into 4 categories
   - One hot encoding of categorical parameters
   - Create Linear regression model. Price as target variable and parameters as independent variables.

4. Scaling the parameter attribute's values within 0 and 1.

5. Importance of parameter calculation using ML algorithm.

6. Final Score table by summing parameter's attribute value.



Business Value:

1. Enhanced Value Discovery: The solution enables buyers to discover the best value for their price range by providing product recommendations with better specifications. Buyers can make more informed decisions and get the most value out of their budget.

2. Market Price Sensitivity: By offering insights into similar or better specification products within the GeM platform, the solution keeps buyers informed about current market trends. This market price sensitivity ensures buyers are aware of their options and can make better purchasing decisions based on comprehensive market knowledge.

3. Improved Transparency and Efficiency in Public Procurement: The solution aligns with GeM's objective of enhancing transparency and efficiency in public procurement. It improves various processes such as: Bid Process (seller evaluation and technical rejection), Incident Management and Catalogue Management System (CMS) which detects price abnormalities and ensures sellers quote realistic prices, maintaining the integrity of the product listings.

III. Government Initiative:

Business Requirement:

GeM wants to build a product recommendation based on government initiative such as Micro & Small Enterprise (MSE) and Make in India (MII).

Today on GeM portal, buyer can shortlist products based on a specific set of parameters and get the best price for the given set of parameters. However, no options / directional recommendations based on the government initiatives are provided to the buyers on possible parameter combination, that is not only an equivalent of the selected set of parameters but also is associated in government driven initiative.

With this solution, buyers will gain access to:

- **Similar MSE Products at Lower Prices**: Identifying products with equivalent features at more competitive prices.
- **Similar MII Products at Lower Price**: Highlighting options that offer superior features or quality without exceeding the buyer's budget.

By implementing the government initiative module using product similarity module, GeM aims to help buyers procure products having required govt initiative feature and can fulfill their quota of procurement mandated by Government of India.

Proposed solution:

Government initiative-based product recommendation module leverage the Product Similarity module to identify similar products with similar or better feature of a product.
In addition to this, the module also filters out the products which qualify to the government driven initiatives and provide that to the current selection by the buyer.

The objective of the Government initiative-based product recommendation module is to identify similar products associated with government driven initiatives for a given product of interest based on three primary criteria:

- The shortlisted products are fulfilling the govt initiative criteria.
- Features of the shortlisted products are similar to the product of interest
- Price of the shortlisted product should be lower than the product of interest / relatively better priced with upgraded components

Solution Consumption:

Currently, this solution is integrated into the Product Display Page (PDP) of the GeM portal, providing buyers with actionable recommendations directly where they view product detail.

Home > Electrical Systems and Lighting and Components and Accessories and Supplies > Lighting Fixtures and Accessories > Interior lighting fixtures and accessories >

White - LED Based Solar Street Lighting System (Q3 Category)

Philips 12 Watt WHITE-LED BASED SOLAR STREET LIGHTING SYSTEM With 30 Ah Battery Capacity

Philips[R]

(BRP710 P LED22 CW 12V Solar P4341)

₹27,000.00 37% OFF

Trends

Product Details

Price For :	1 pieces
MRP/Unit:	~~₹ 43,000.00~~
Offer Price/Unit:	**₹ 27,000.00**
Availability:	200 In Stock
Min. Qty. Per Consignee:	5
Product id:	5116877-11791493956
Country Of Origin:	India
Local Content (MII):	Not Declared

Specifications

Minimum Power output of PV Module	75 Wp
PV Module type	Crystalline Silicon Terrestrial Photovoltaic (PV) modules
Minimum effieiency of PV Module (in %)	14
Battery voltage, Minimum (in Volts)	12.8
Battery capacity, Minimum (in Ah)	30
Battery type	Lithium Ferro Phosphate
Battery enclosure	Builtin Luminaire with IP65 protection
Rating	12 Watt
Total luminous flux	≥ 1500 lm.
Luminous efficacy (i.e. system efficacy)	≥ 125 lm/W.
Color Temperature	5500 K to 6500 K
Overall total Efficiency of the Electronics	>/=90%
Ingress Protection (IP) for Optical and Control gear compartment	IP 65

VIEW LESS

Similar MII Products at GeM

Unbranded 12 Watt WHITE LED BASED SOL...

Brand: NA
Min. Qty. Per Consignee: 2

₹ 12,000.00 90% OFF
~~₹ 120,000.00~~

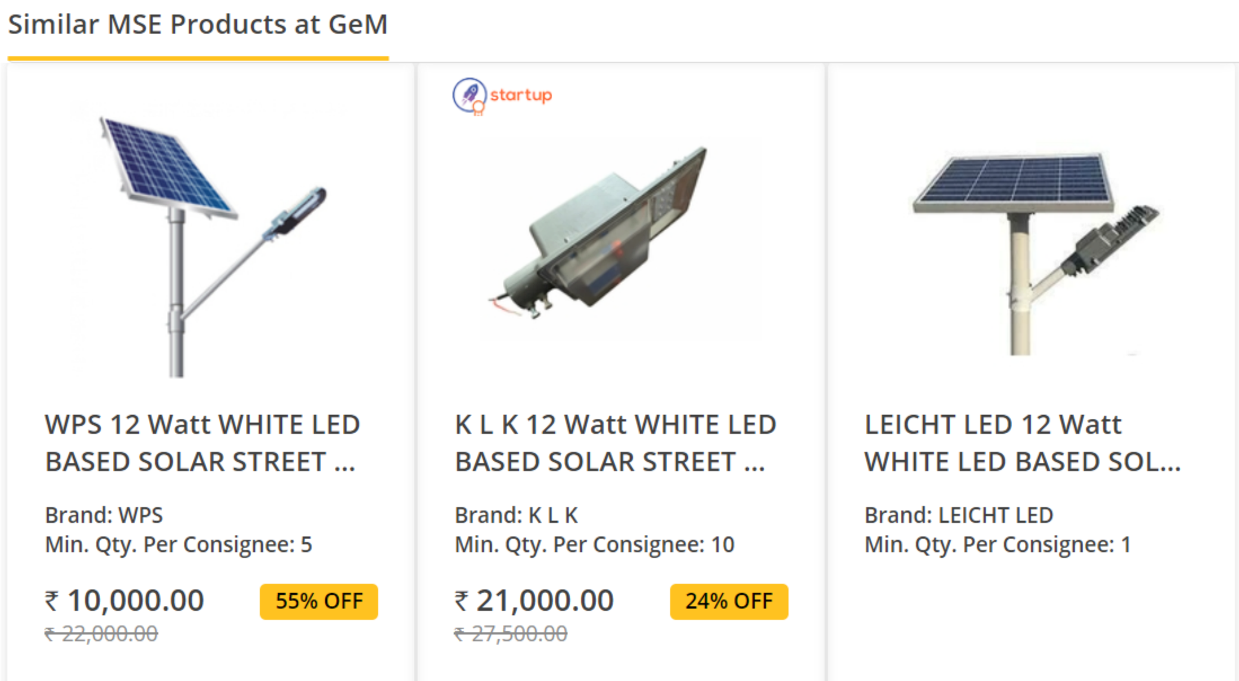
Unbranded 12 Watt WHITE LED BASED SOL...

Brand: NA
Min. Qty. Per Consignee: 5

₹ 14,500.00 48% OFF
~~₹ 28,000.00~~

ENERGY EXPERT 12 Watt WHITE LED BASED SOL...

Brand: ENERGY EXPERT
Min. Qty. Per Consignee: 1



Data:

To build this solution, we will be using data from product similarity module as well as some other tables which has the information regarding the govt initiatives. In future there may some table will be added or removed as per the information given by GeM task force. Apart from all the tables required for product similarity, we will be needing below mentioned tables:

- Table: inb_seller
- Table: arx_tb_arm_branch
- Table: arx_tb_arm_user_entity
- Table: arx_tb_arm_master
- Table: arx_tb_arm_branch_env

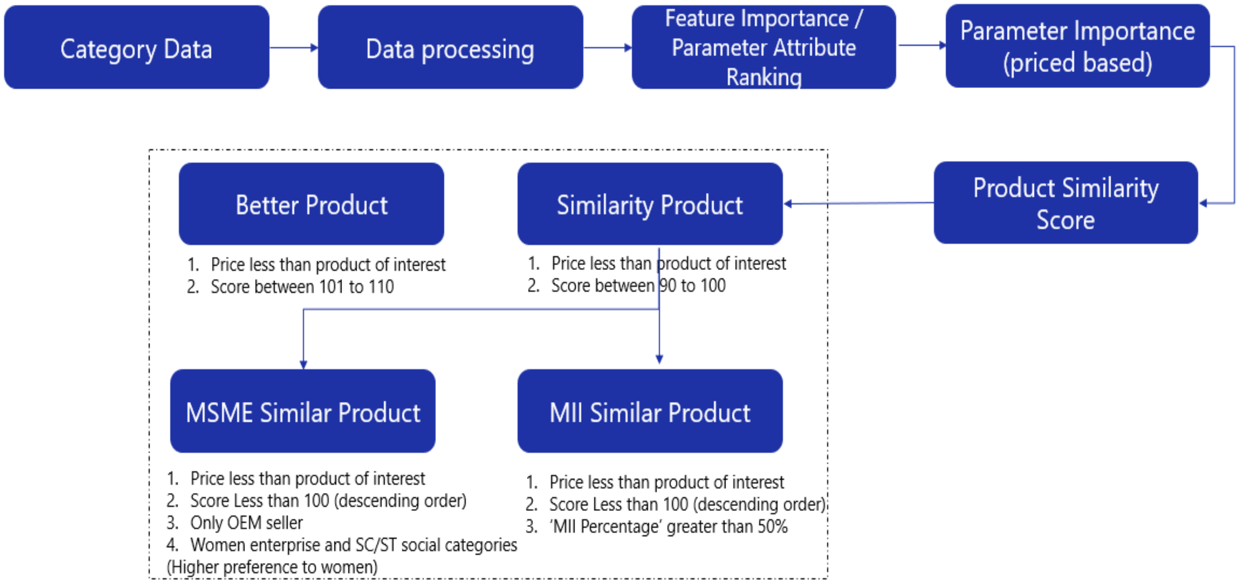
Model Methodology:

The module works on the outcome of the product similarity module to determine which products are similar to the selected product based on the specifications. Over the selected products an additional filter is added to determine similar products associated with government initiative.

Steps In Model:

1. Find list of products fulfilling govt initiative.

2. Run Product Similarity module to identify most similar product from the above mentioned list.



Business Value:

Qualitative – Buyers will get recommendation to fulfill their quota and sellers will be encouraged to register under government driven initiatives.

IV. PGA – Abrupt:

Business Requirement:

In GeM portal, buyer can shortlist products based on a specific set of parameters and get the best price for the given set of parameters. However, no options / directional recommendations are provided to the buyers about prices which are too highly quoted by buyers.

Presently, the GeM portal focuses solely on best price discovery rather than discovering the best value for the price. This means buyers are not informed about the prices of a particular product as well as the prices of similar products. As a result, buyers may not be making the most informed purchasing decisions and might be paying more than necessary.

To address this gap, GeM wants to develop price gap analysis to assist buyers. This module will offer a list of prices within the range as well as prices beyond the threshold which is considered as abrupt price. By providing these recommendations, buyers can make more informed decisions, ensuring they receive the best value for their money.

This solution will help in identifying products with equivalent features at abrupt price. By implementing the price gap analysis-abrupt, GeM aims to enhance the purchasing experience, ensuring buyers achieve optimal value for their expenditure.

Proposed solution:

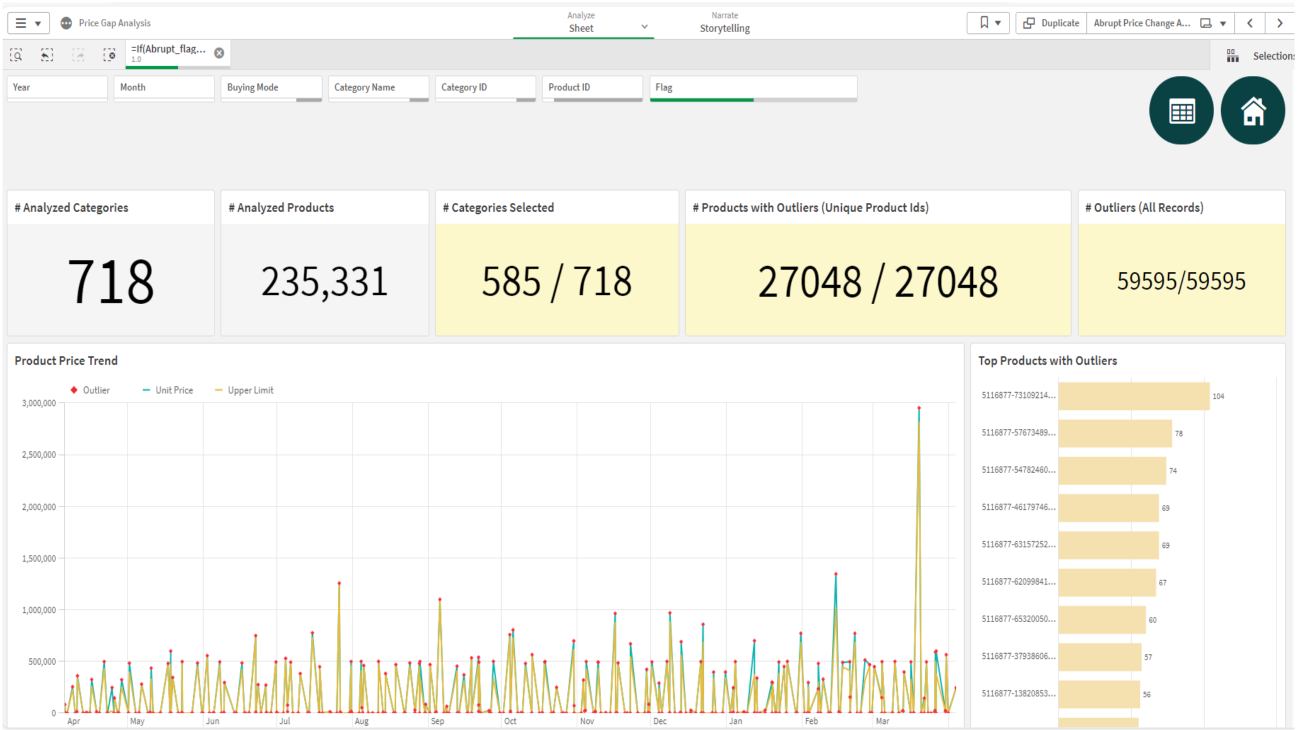
The objective of this solution is to identify the products which are priced higher. The Price Gap module will transform the GeM portal from a platform focused solely on price discovery to one that also emphasizes value discovery.

By offering intelligent, machine-learning-driven product comparisons, buyers will have access to a wider range of options that meet their needs both in terms of features and cost, ultimately enhancing their overall purchasing experience on the GeM portal.

Solution Consumption:

This solution is available through BI dashboard built in Qlik application as showcased below.

It gets monthly updated and accessible to GeM stakeholders for monitoring and to take appropriate actions on anomalous price behavior.



Data:

To build this solution, required data consist of all the product id's in a category, along with buyer id, buyer name, seller name, value, quantity etc. The required data can be fetched from below tables.

- Table: ffm_order
- Table: ffm_orderitem
- Table: ffm_buyer

Model Methodology:

This solution is dependent on product similarity solution and its outcome utilized in this solution. The historical prices of set of similar products are analyzed and a threshold identified for group of similar products using Mahalanobis outlier method. This threshold helps to detect outlier prices i.e. abrupt prices.

Steps In Model:

Below are the steps in the model development.

1. Extract the required data:
   - From 'ffm_order and 'ffm_orderitem table, we get the seller-id, quantity, buying mode, sold price etc.,
   - For ffm_ buyer, we get buyer details like buyer name, buyer state, buyer organization.

2. Clean data (pre-processing)
    - Remove values like “OrderCancelled”, “CancellationRequestedByBuyer”,” OrderRejected” in column called ‘wfstatus’.
    - Rename the parameters

3. Feature Engineering:
    - Fetch score column from the output of product similarity data and merge it in the price analysis data.

4. Grouping the Data:
    - The dataset is grouped by Group_id and category_id using the pandas .groupby() method. This step ensures that each unique combination of Group_id and category_id forms a distinct group, on which further analysis is performed.

5. Data Point Check:
    - For each group, we check the number of data points. If a group contains five or more data points, we proceed to the next steps. Groups with fewer than five data points are not analyzed further, this way it ensures that a statistically significant sample size is maintained for each group.

6. Initialization of the Flag Column:
    - A new column, flag, is initialized for each group, with an initial value of 0. This column will be used to mark outliers in the dataset.

7. Cutoff Calculation:
    - For each group, the mean and variance of the unit_price column are calculated. Next step is to check if the variance is non-zero. A variance of zero indicates that all the unit prices in the group are identical, making it impossible to compute meaningful Mahalanobis distances. If the variance is zero, the group is skipped. Finally The Mahalanobis cutoff is calculated using python function. This value serves as a reference point to help identify extreme outliers in the price distribution.

8. Outlier Detection:
    - Any data point with a Mahalanobis distance greater than this threshold is flagged as an outlier.

9. Updating the Flag Column:
    - For data points identified as outliers, the corresponding entries in the flag column are updated to 1, indicating that the unit price for these entries is an outlier.

10. Appending Group Results:
    - Once the outlier detection process is completed for a group, the modified group data (with flags and cutoff prices) is appended to the final DataFrame. This process repeats for each group in the dataset.

11. Final DataFrame:
    - After processing all groups, the final DataFrame contains the original data, along with two new columns: flag (indicating outliers) and cutoff_unit_price (indicating the unit price cutoff for detecting outliers in each group).

Business Value:

Control high MRP & high offer price being defined by Sellers for Risk Mitigation and Fraud Prevention Outlier price detection helps identify unusually high or low prices, which could signal potential risks such as pricing errors or fraudulent activities. By flagging such anomalies, GeM can take corrective action. For instance, unusually high prices could be a sign of manipulative practices by suppliers or vendors. This helps GeM for maintaining pricing integrity and transparency, reducing the risk of customer dissatisfaction.

V. PGA – Internal

Business Requirement:

In the GeM portal, buyers can currently shortlist products based on specific parameters to discover the best price available. However, the portal does not provide a method for validating whether the price of the selected product aligns with similar products, potentially leading to decisions based on incomplete information.

Currently, the GeM portal focuses on price discovery but does not provide a mechanism to ensure that buyers are paying a fair price in comparison to other similar products. This lack of visibility into price variations among similar items could result in buyers paying more than necessary for a product.

To address this gap, GeM aims to develop a Price Gap Analysis module for internal price validation. This module will allow buyers to validate the price of a selected product against a list of similar products identified using a product similarity solution. Key features include:

- Price Similarity Validation: Identifying similar products that have the same score as the selected product based on features and characteristics. Here product similarity solution's outcome is utilized to perform price similarity validation on similar products.
- Minimum and Maximum Price Thresholds: Using statistical techniques, the solution will provide price variation thresholds based on historical sold prices and offer prices for similar products in CMS.

By implementing the Price Gap Analysis for internal validation, GeM seeks to provide buyers with the tools to make more informed decisions, ensuring that their selected product's price is within a fair and reasonable range, ultimately improving value for money.

Proposed Solution:

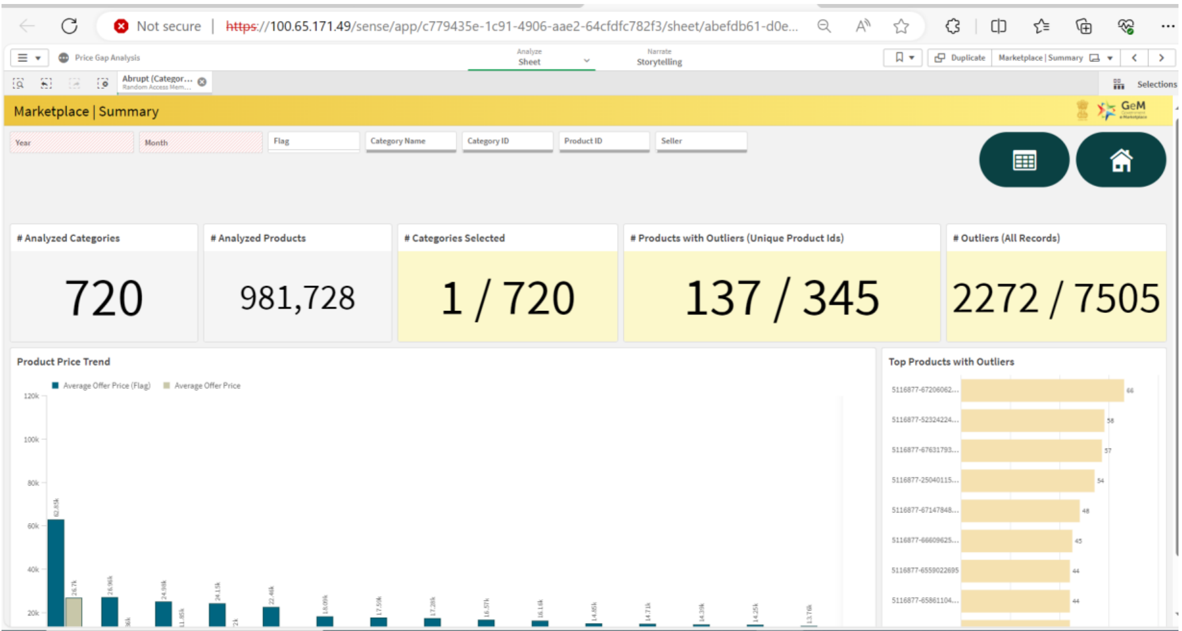
The Price Gap Analysis – Internal/CMS module is an and statistical-powered system designed to validate the price of a selected product by comparing it with a list of similar products that have the same score based on specific features.

The objective of the price gap analysis – internal is to provide buyers with a detailed understanding of price variations among similar products by employing minimum and maximum thresholds. This Price Gap module will transform the GeM portal from a platform focused solely on price discovery to one that also offers validation of pricing fairness.

The solution will allow buyers to see price ranges based on historical sold prices and offer prices. This will empower them to make more informed decisions, ensuring they are paying a fair price within the context of the marketplace, and ultimately enhancing the overall purchasing experience on the GeM portal.

Solution Consumption:

This solution is deployed on GeM server. And available through dashboards and CMS API (currently in testing phase). GeM validation team has the access to dashboard to monitor and take appropriate actions.



Data:

To build this solution, required data consist of sold price data and offer price data for all the category id's. The required data can be fetched from below tables.

- Table: inb_variants
- Table: inb_stock
- Table: inb_browse_nodes
- Table: inb_sellers
- Table: inb_catalog_attrs

Model Methodology:

1. Following four ways the data finalized to define the thresholds:
    - For any selected product, the first step is to check its historical sold price transactions. A minimum of five transactions is required for that product to ensure sufficient data for analysis.

    - If this is not possible due to an insufficient number of transactions, the next step is to check the sold price transaction data for similar products within the same category or group. Again, a minimum of five transactions is necessary for this group of similar products to proceed with the analysis.

- If both of the above conditions cannot be met, the next option is to use the offer price data for the selected product. In this case, at least five different sellers must have quoted prices for that product on the marketplace to ensure a reasonable comparison.

- If none of these scenarios provide the required amount of data, the system will indicate that it cannot perform a price sanity check due to insufficient information.

2. Once the required data is available from one of these steps, the next task is to calculate the price thresholds. The minimum price and maximum price are identified from the available data.

3. The minimum threshold is calculated as 50% of the minimum sold price (Min Threshold = 0.5 * min sold price), while the maximum threshold is set at 1.5 times the maximum sold price (Max Threshold = 1.5 * max sold price).

4. These thresholds provide a range for assessing whether a price is fair, too low, or too high based on historical data or offer prices.

Note: PGA abrupt data will be used as input for sold price analysis and removed outliers before proceeding to marketplace analysis.
Historical time period for sold data – 1 year

Steps In Model:

1. Extract the required data:
   - From inb_variants , we get the make, model, title etc. of a product.
   - For inb_sellers, we get seller id and we will be getting the catalog id from inb_catalog_attrs.

2. Clean data (pre-processing):
   - Read the output of abrupt use case and remove the flagged products.

3. Transaction-Based Price Analysis – Sold Price Analysis:
   - The code begins by calculating the transaction count for each product using historical sold price data. Products with fewer than five transactions are separated for further analysis, while products with five or more transactions proceed to calculate the minimum and maximum price ranges. For products with insufficient transactions, the code checks if similar products within the same group have enough sold price data (at least five transactions). If sufficient group data is found, the minimum and maximum price ranges are computed; otherwise, the product is flagged for further investigation.

4. Offer Price-Based Price Analysis:
   - If a product does not have enough historical sold price data, the code then checks offer prices from the marketplace. It calculates the minimum and maximum offer price ranges for products that have quotes from at least five different sellers.
   - If the criteria are not met, the system flags the product as having insufficient information to determine price ranges. This ensures that the system can handle cases where transaction data is limited by falling back on offer price analysis.

5. Final Price Range Determination:
   - The final step involves determining the most appropriate price range for each product, using either sold price data or offer price data, depending on which is available. The system assigns the minimum and maximum price ranges and the corresponding status to each product, ensuring that valid data is used. If both sold price and offer price data are unavailable, the product is marked as "cannot perform sanity due to insufficient information."

6. Handling Special Cases (Price Below 5 Rupees):
   - The code handles special cases where the product's sold or offer price is less than 5 rupees. If the offer price for any product is below this threshold, the code recalculates the minimum price and updates the final price range accordingly. The final results are then saved to a CSV file, ensuring the correct data is captured and output for further analysis.

Business Value:

1. Enhanced Price Validation and Purchase Decision-Making:
   By leveraging the minimum, maximum, and median prices of selected products and their similar counterparts, buyers within the GeM portal can make more informed and validated purchasing decisions. Understanding the price range for a product, along with comparable alternatives, empowers buyers to avoid overpaying or selecting products quoted at abnormally high prices. This not only optimizes the purchasing process but also ensures that buyers maximize value for their expenditure by benchmarking products against comparable offerings within GeM.

2. Risk Mitigation and Efficient Procurement:
   With access to comprehensive price ranges, buyers can better identify pricing anomalies, whether excessively high or low, and take corrective actions during procurement. This aids in mitigating risks such as over-inflated prices and ensures transparency in vendor pricing. Additionally, this price validation process improves procurement efficiency, as buyers can avoid engaging with suppliers who may be quoting prices significantly outside the norm.

VI. PGA - Min Max

Business Requirement:

This solution provides buyers trend of price of the product using their minimum and maximum price for time periods. This enables buyer to take decision of bulk procurement.

GeM aims to develop a Price Gap Analysis module focused on both the min-max pricing strategy. This module will provide buyers with comprehensive insights into price ranges, including:

- Minimum and Maximum Prices: Identifying the lowest and highest prices for a specific product and similar products.
- Median Price: Determining the median price for a product and similar products, offering a balanced view of pricing trends.

By implementing the Price Gap Analysis for min-max pricing, GeM intends to enhance the purchasing experience for buyers, ensuring they can make informed decisions and achieve optimal value for their expenditure

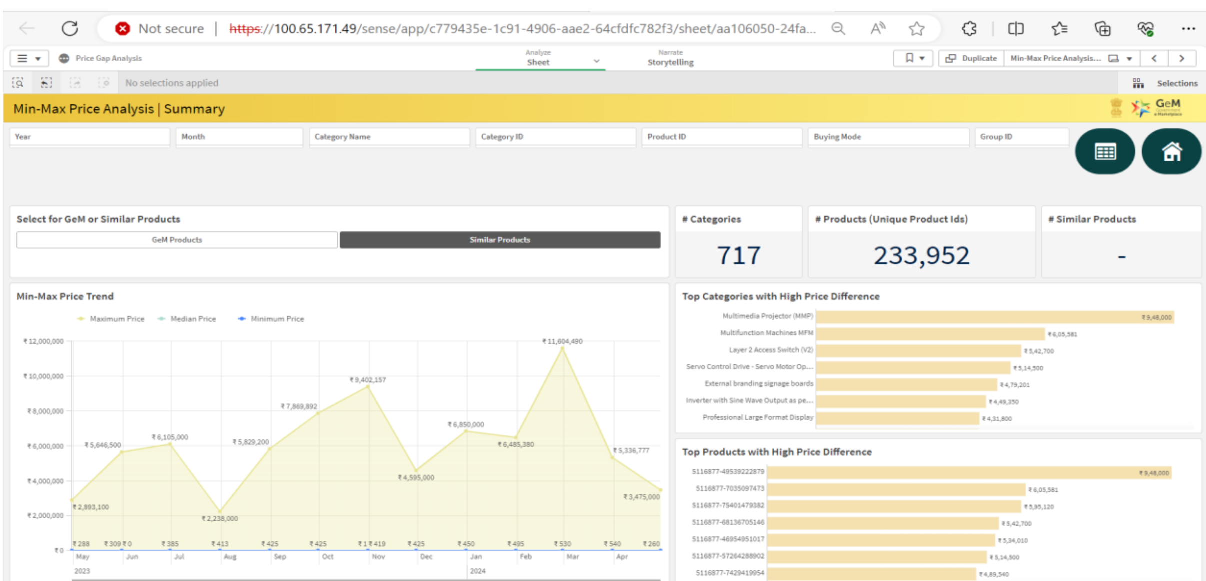
Proposed Solution:

This solution helps buyers to provide trend analysis of min and max prices of products and their similar products. For a product id, its minimum and maximum prices can be extracted across time period.

Median of the price can be compared with minimum and maximum trend.

Solution Consumption:

This solution is available through BI dashboard built in Qlik application as showcased below.
It gets monthly updated and accessible to GeM stakeholders.



Data:

To build this solution, required data consist of all the product id's in a category, along with buyer id, buyer name, seller name, value, quantity etc.. The required data can be fetched from below tables.

- Table: ffm_order
- Table: ffm_orderitem
- Table: ffm_buyer

Model Methodology:

This solution identifies minimum and maximum prices for a product and its similar product from historical prices and provides price trend. Result of product similarity is consumed to get similar products and their associated product groups.

Steps In solution:

1. Extract the required data:
    - From 'ffm_order and 'ffm_orderitem table, we get the seller-id, quantity, buying mode, sold price etc.,
    - For ffm_ buyer, we get buyer details like buyer name, buyer state, buyer organization.

2. Clean data (pre-processing)
    - Remove values like "OrderCancelled", "CancellationRequestedByBuyer"," OrderRejected" in column called 'wfstatus'.
    - Rename the parameters

3. Feature Engineering:
    - Fetch score column from the output of product similarity data and merge it in the price analysis data.

4. Data Filtering and Preparation:
    - The first step involves filtering the data to extract rows corresponding to a specific category using. The resulting filtered DataFrame is then sorted by 'order_date' to ensure chronological order.
    - The 'order_date' column is converted to datetime format to facilitate time-based grouping, and the median of 'unit_price' is calculated to serve as a benchmark price for comparison during price analysis.

5. Grouping and Aggregation:
    - Next, the data is grouped by 'Product_id' and 'order_date' at a monthly frequency. For each product, we calculate the minimum price (min_price), maximum price (max_price), number of units sold (units_sold), and create a list of prices within that period (prices_list).
    - This allows for a detailed analysis of the price trends and distribution. Any rows with missing values are removed to ensure data integrity, and the 'category_id' is retained by extracting it from the first row of the filtered DataFrame.

6. Merging and Finalizing the Output:
    - In the final step, the aggregated data is merged with another dataframe on 'Product_id' to obtain additional attributes such as 'Group id'.
    - This merged DataFrame, contains all necessary information for further analysis, including the min/max prices, units sold, and the benchmark/median price.

Business Value:

Informed Pricing Decisions and Competitive Benchmarking: By identifying the minimum, maximum, and median prices of products and their similar counterparts, GeM buyers can make more informed pricing decisions. Similarly, knowing the price range of similar products enables benchmarking within a category.

VII. Demand Forecasting:

Business Requirement:

Build, evaluate and deploy time-series based models (Autoregressive Integrated Moving Average (ARIMA), Seasonal ARIMA for seasonal products/services and Exponential Smoothing) for forecasting products / service demand and pricing with historical transactions data.

Proposed solution:

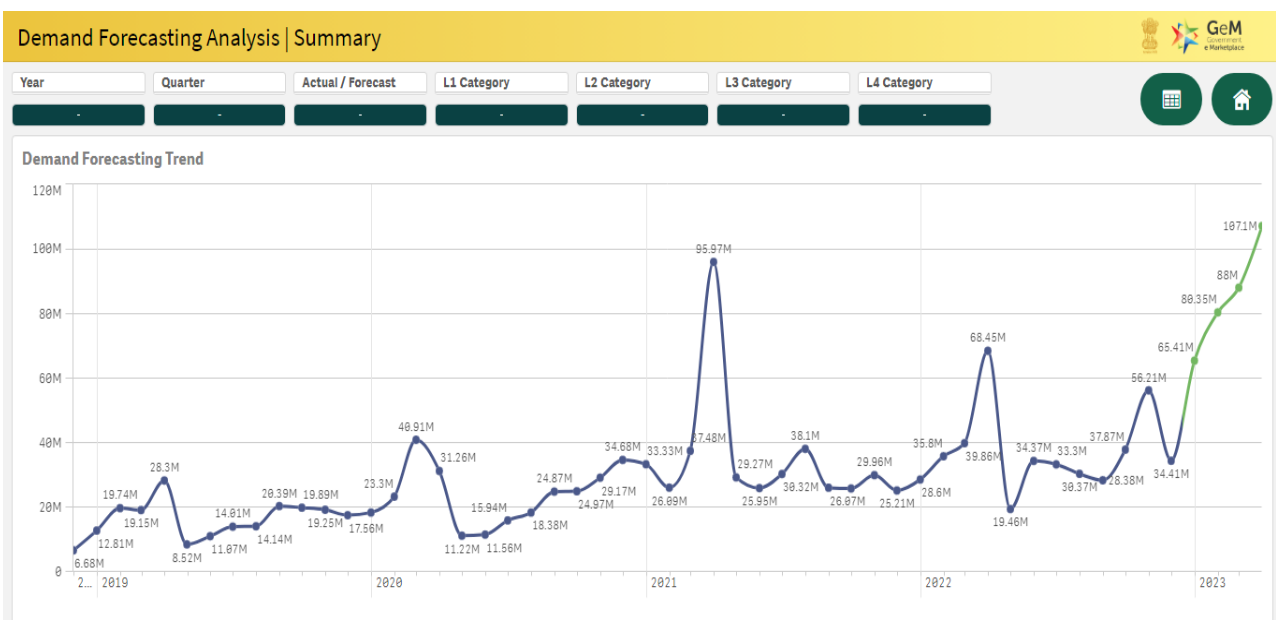
Business can leverage demand forecasting to enhance inventory management by anticipating future sales based on historical sales data and strategic planning. This approach aids in making informed decisions on inventory planning, storage needs, executing flash deals, and meeting consumer expectations.

Forecasting demand for future timelines allows sellers to accumulate sufficient supplies in anticipation of demand surges, enabling them to participate and fulfill orders on time

Solution Consumption:

The Qlik dashboards are designed to display demand forecasting for L1, L2, L3, and L4 categories. Users can select specific categories to view historical quantity data up to the last month and forecasted values for the next 12 months.

These dashboards are updated monthly to ensure they reflect the most current data.



Data:

Historical purchase quantities for each category dating back to November 2018.

Model Methodology:

1. The module operates by aggregating the demand for a category over a specified time period (monthly), pre-processing the data, and then forecasting future demand.

2. Time series forecasting is employed to make scientific predictions based on historical time-stamped data. This process involves building models through historical analysis and utilizing them to generate insights that inform future strategic decision-making

   - Methodology1: (Shipment Date)
     Use the `created_at` column as the delivery date from the `inb_prcs` table. If `created_at` is missing in the `inb_prcs`, exclude those specific orders from consideration.

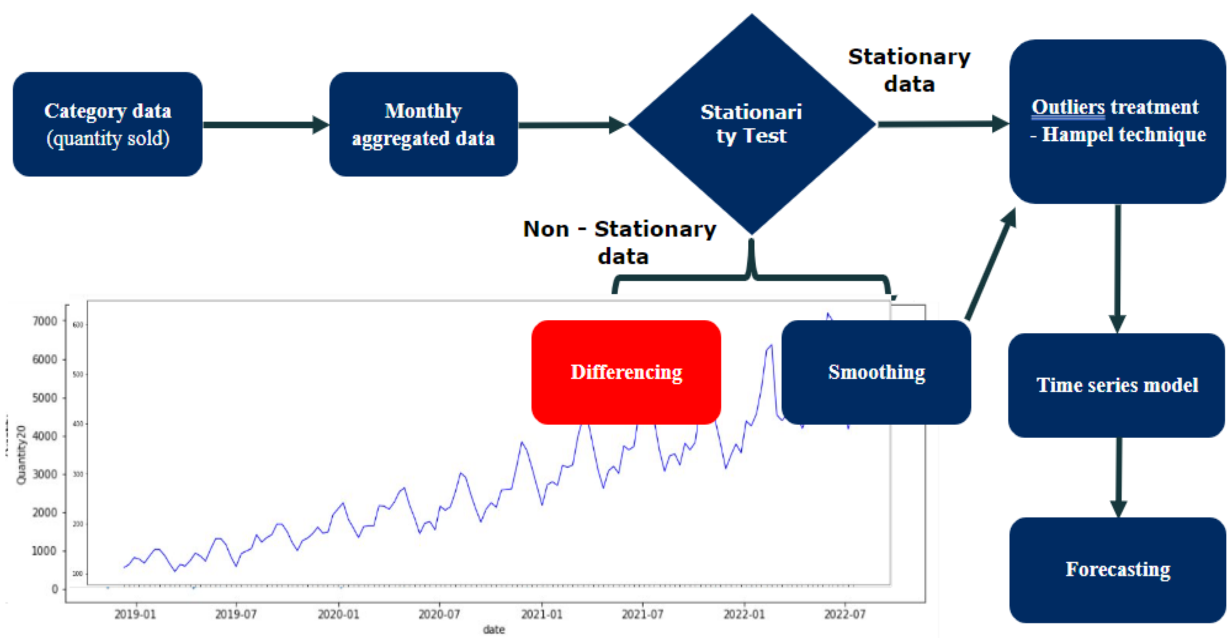
   - Methodology2: (Order Date)
     Utilize the `confirmed_at` column as the order date from the `ffm_order` table and the `quantity` column from the `ffm_orderitem` table. This approach will be applied for both monthly and quarterly predictions.

Steps In Model:

1. Only consider the categories having at least 40 months data points

2. Not to consider the categories having continuous 4 months missing data

3. Some contract numbers are excluded (which were shared by GeM)



The model performance is evaluated on these crucial metrics:

- MAE (Mean Absolute Error)
- RMSE (Root Mean Squared Error)
- MAPE (Mean Absolute Percentage Error)
- SMAPE (Symmetric Mean Absolute Percentage Error)
- MSE (Mean Squared Error)
- MAD (Mean Absolute Deviation)
- R-squared (Coefficient of Determination)
- EV (Explained Variance Score)
- MDA (Mean Directional Accuracy)
- WMAPE (Weighted Mean Absolute Percentage Error)

Business Value:

1. Qualitative – Sellers will get the tentative quantity required/demanded and can prepare for it in required time. This will enable more participation and generate a healthy competition in GEM.

2. Buyers can also plan their procurement effectively by knowing in advance about seller's availability.

VIII. Buyer Seller Collusion:

Business Requirement:

Detect and report patterns of one buyer buying only from certain sellers and details around those purchase. To develop a methodology which can identify buyer seller collusions.

Proposed solution:

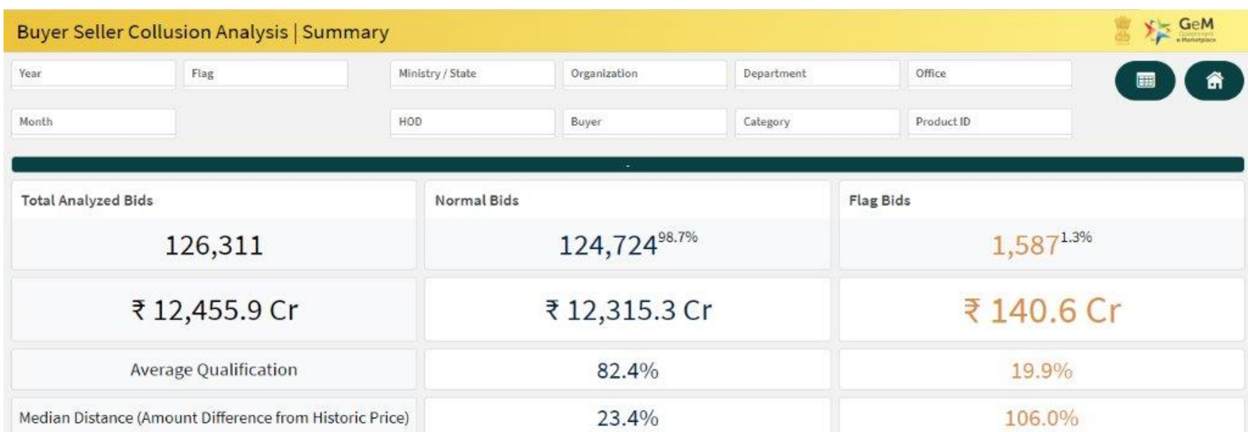
Instead of establishing predefined business rules, we should allow the data to define the criteria for identifying collusions. By analyzing data at the category level, we can capture the inherent behavior within each category, setting a benchmark for what constitutes an anomaly. This approach involves considering buyer-related data such as purchase frequency within a category, price paid, historical pricing, and repeated transactions with the same buyer.

From the seller's perspective, we examine factors like price trends for the specific product and price comparisons with other (re)sellers offering the same catalog specifications.

Solution Consumption:

The solution is integrated into Qlik Dashboards, highlighting potential collusion cases. The GeM team can then review these flagged bids and notify the respective buyers and sellers if necessary.

The dashboard is updated monthly with the latest data.



Data:
To build this solution, required data consist of

- Buyer Details
- Seller Details
- Product / Variant Details
- Bid Details
- Transaction Details

Model Methodology:

1. Bid Details:

# of participants : P
# of qualified sellers : Q
Qualification ratio : P/Q
L1 price of the BID : L1

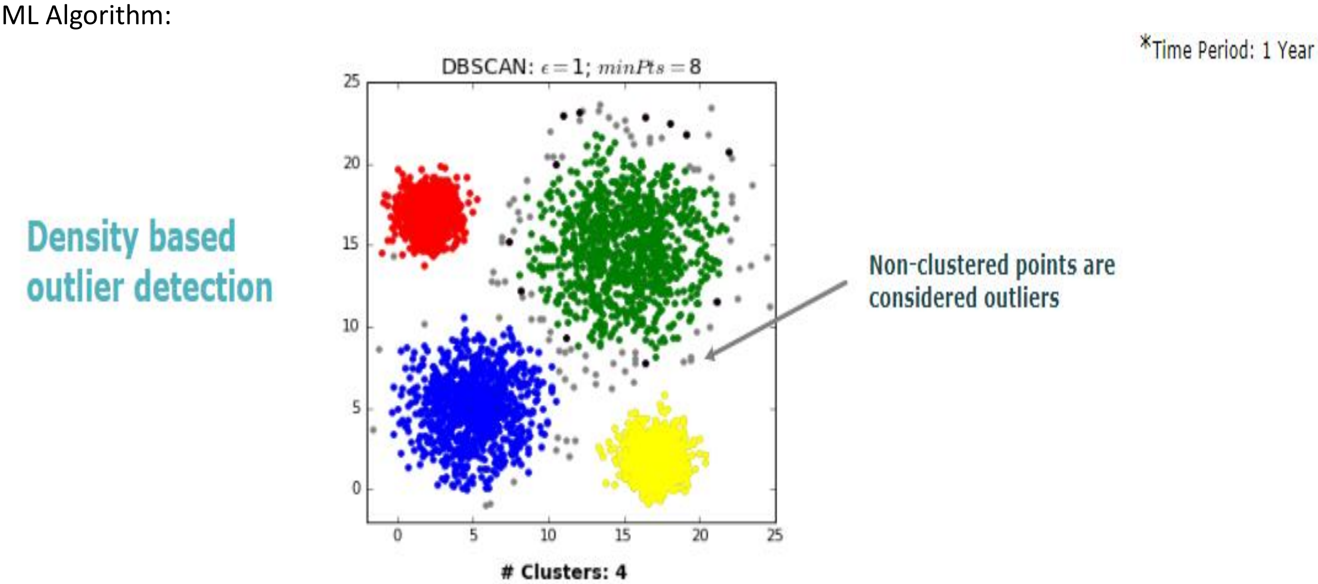
2. Product details:

Mean/Median of the Product : M
Price difference from Mean/Median : L1 - M
%Price difference from Mean/Median : (L1 – M)/M

3. Buyer/Seller details:

Total number of orders
Number of orders in a category
Number of orders between buyer-Seller
Number of orders between buyer-Seller in a category
Historic offering at better price

ML Algorithm:



Steps In Model:

1. Data pre-processing: Generate the following external variables for further analysis:
   - Percentage mean/median distance – Calculate the percentage mean/median distance from the lowest price quoted in the bid.
   - pt_ratio - Determine the ratio of the number of qualified sellers to the total number of sellers participating in the bid.
   - sb_trans_ratio - Calculate the ratio of the number of orders between a specific seller and buyer to the total number of orders placed by that buyer across categories.

2. Model Training: Train the mixture model to identify potential cases of collusion.

3. Final Report: Compile a report detailing pairs of buyers and sellers who may be colluding within a given product category.

To reduce false positives and minimize the number of results requiring validation, the following rules are recommended for identifying outliers:

- The number of qualified sellers should be fewer than 5
- The Qualified Ratio should be less than 30 %
- The Median Distance should be greater than 15%

Business Value:

1. Ensures the GeM portal operates at peak efficiency, maintaining the integrity of free competition within the platform.

2. Fosters a competitive and equitable marketplace for both buyers and sellers.

IX. Buyer Seller IP Collusion:

Business Requirement:

One of the potential collusion scenarios, as defined by a business rule, could involve both the buyer and seller, or multiple sellers, participating in a procurement process from the same location or network, identified by having the same IP address. This could indicate collusion aimed at winning the order, thereby undermining the principle of promoting fair competition.

Proposed solution:

The solution identifies instances where both buyers and sellers, or multiple sellers, have used the same IP address while creating or participating in a bid. This scenario could indicate potential buyer-seller or seller-seller collusion, suggesting attempts to manipulate the procurement process.

Solution Consumption:

The Qlik dashboard is designed to track bids where buyer and seller participated in bids IP address are matched. GeM SPV can send mail notification to HOD for these bids with same seller and buyer IP address along with seller and contract information.

Data:
To build this solution, required data consist of:

- Buyer Details
- Seller Details
- Bid Details

Model Methodology:

For this a business rule is set, that compares the IP addresses used by the seller for the order with the IP address of the buyer or any other seller while completing the procuring.

We leverage the basic fact of that the IP addresses of the users are captured in multiple occasions while participating in any action of procurement. These events are mapped together to build relationship among buyers and seller and the pair/s with same IP addresses are concluded as potential collusion.

Steps In Model:

Image: process flow

Business Value:

1. Qualitative: This solution ensures the GeM portal operates to its fullest potential by maintaining the integrity of the procurement process. It helps prevent transactions that could undermine the spirit of fair competition within the GeM platform.

2. Quantitative: By capturing and mapping IP addresses, GeM can effectively identify and establish relationships between sellers and buyers, providing a valuable tool for detecting potential collusion and ensuring transparency in transactions.

X. Intentional Hiding:

Business Requirement:

The GeM marketplace aims to ensure that all products listed are easily discoverable by buyers. However, sellers may attempt to hide certain products intentionally, either within the same or different categories, by providing unique or incorrect specifications.

This requirement focuses on detecting such intentionally hidden products to ensure transparency and enhance the quality of listings.

Proposed solution:

The proposed solution aims to identify hidden product catalogs by analyzing product specifications within categories, flagging entries with unique or incorrect attributes, and exporting the results for marketplace administrators to review.

This will ensure that hidden products can be detected and corrected, improving searchability and reducing potential misuse of the system.

Solution Consumption:

The proposed solution can be effectively consumed by two key stakeholders:

- Seller: Sellers will be restricted during the catalog upload process in CMS. The system will automatically check the product specifications entered by sellers, and if they match any criteria of hidden or incorrect specifications (e.g., unique or uncommon attributes), the upload will either be flagged or restricted.

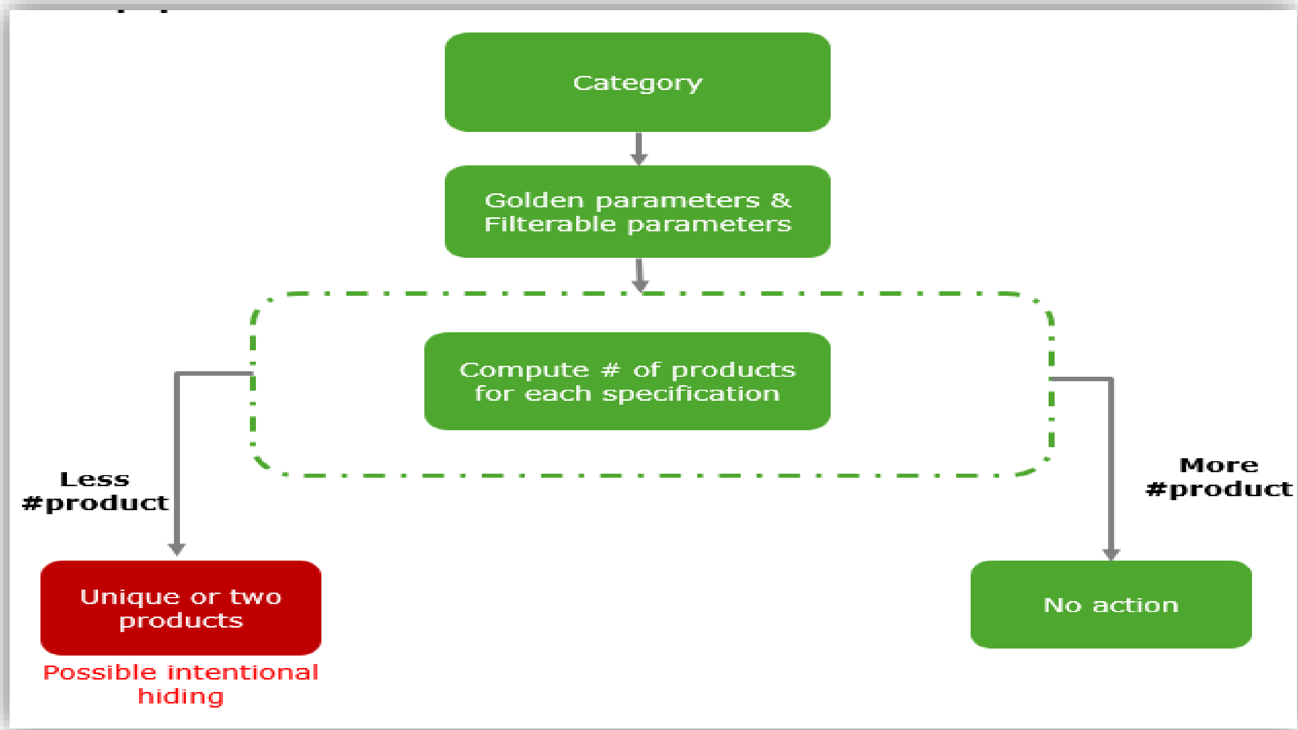
- GeM SPV: GeM SPV can utilize the flagged product data to perform market sanitization by reviewing products with rare or suspicious specifications. The solution allows GeM administrators to proactively monitor and take action on these flagged products.

Data:

The required data can be fetched from below tables to identify intentionally hidden product catalogs in the GeM marketplace:

- Table: inb_variants
- Table: inb_catalog_attrs

Model Methodology:



Steps In Model:

1. Data Extraction:
   - Source: Product information will be extracted from the GeM staging database using SQL queries. This includes product ID, category information, catalog ID, and technical parameters.
   - Target: The focus will be on active products within specified categories.

2. Technical Parameter Cleaning:
   The raw product technical parameters often contain unnecessary lines (e.g., weights) and formatting issues. These will be cleaned using a Python function to remove irrelevant data and standardize the parameters for further analysis.

3. Mapping Specifications to Categories:
   Using a predefined category specification mapping file, the technical parameters will be mapped to human-readable names for each category. This step ensures that all product attributes can be easily interpreted and compared.

4. Concatenation of Product Specifications:
   All relevant technical parameters for a product will be concatenated into a single string. This allows for easy identification of unique or rare combinations of specifications that might indicate hidden products.

5. Frequency Analysis of Specifications:
   Once specifications are concatenated, the frequency of each unique combination will be analyzed within the same category. Products with specifications that occur less than a certain threshold (e.g., three or more occurrences) will be flagged as potentially hidden or incorrectly categorized.

6. Flagging Hidden Products:
   Products with low-frequency specification combinations will be flagged for further review. These flagged products may be intentionally hidden or the result of data entry mistakes.

7. Data Export:
   The final dataset, containing product details, concatenated specifications, frequency counts, and flag indicators, will be exported to a CSV file. This file will allow GeM to review and take action on potentially hidden or miscategorized products.

Business Value:

1. Transparency: Ensures that sellers cannot hide products in the marketplace by using incorrect or unique specifications.

2. Error Correction: Identifies and corrects unintentional errors due to misinformed sellers.

3. Improved Searchability: Helps users find relevant products more efficiently by minimizing hidden or misplaced product listings.

4. Enhanced Monitoring: Provides marketplace administrators with a tool to monitor and review flagged product listings for policy violations or errors.

XI. Avoiding Order Aggregation:

Business Requirement:

Currently the buyer visits the marketplace to select the product of interest in order to add it to the cart and initiate the procurement process. The product selected can be procured directly from marketplace if it falls under a specified limit. Any procurement that breach this limit has to be done under the bidding process. To avoid this buyer goes with multiple direct purchases. The bid of more value requires an approval from the higher authority, to bypass this multiple bids with smaller quantity is placed.

Avoiding order aggregation is the method using which the buyer is splitting a larger order to multiple small orders. GeM wants to identify and alert the buyers who are avoiding order aggregation. Orders splitting can be for done in the following ways:

- Splitting an order of potential bid into multiple direct procurement; this is the case where the requirement of solution to detect such cases.
- Splitting an order of potentially high bid into similar smaller bids; not in the scope of this solution.

The splitting of the bid into multiple direct procurement can be done to avoid the long processing period involved in a bidding process or to procure the same product/services for different consignees. The act of splitting the order because of the above reason might save some time but eliminates the competition (in case of purchase under 25K) and best price discovery possible in bidding. The splitting of a bid into multiple bids with smaller value is done to bypass the requirement of permission from superior financial authority. This can allow a buyer to procure a product above the authorized limit and bypass his superiors.

Requirement as per RFP: Detect and Report patterns to identify large orders break down into several smaller portions and buying repeatedly to avoid tendering.

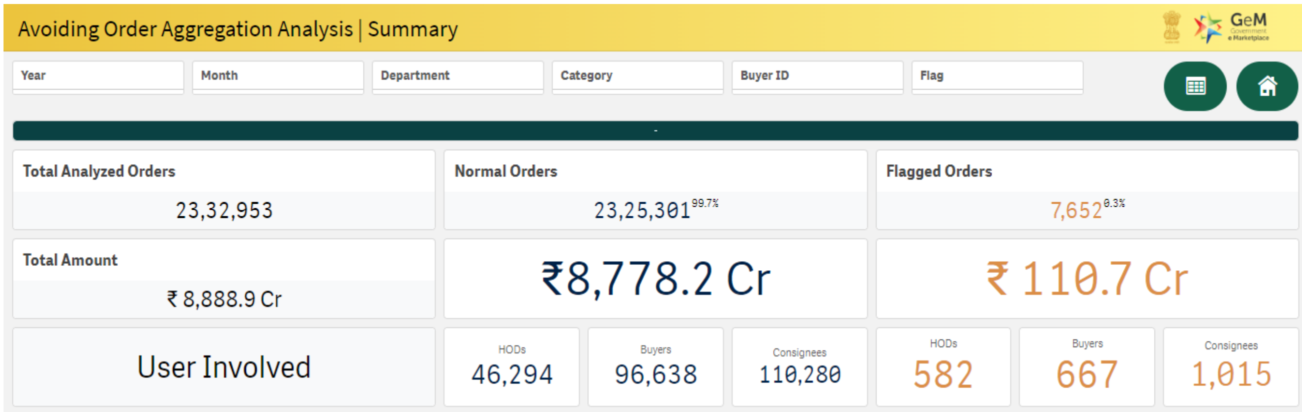
Proposed solution:

Similar to Buyer Seller Collusion, idea is to let data define the rules of order aggregation. Different levels of data like inter-purchase interval (time between two purchases in the same category), % of transactions that are non-competitive (either non-L1 or higher priced than historical transactions), # of unique sellers being transacted with, etc. will be used together to set thresholds based on data to define buyer behavior that considered as avoiding order aggregation.

The monthly report of suspected buyers shared to GeM SPV and by taking corrective action against suspected buyers leads to increase order aggregation, competitive price for the products and reduce the chances of seller buyer collusion.

Solution Consumption:

Currently, this solution is consumed through BI dashboards which is accessible GeM team. Dashboard is updated at monthly level having a report of suspected buyers along with transaction, category and seller details. Below is the screenshot of dashboard:



Data:

To build this solution, data is extracted from following database table

- ffm_order
- ffm_orderitem
- ffm_buyer
- ffm_consignee

And the required data is as below

- order_id
- category_id
- product_id
- product_name
- quantity
- unit_price
- order_amount
- buyer_id
- seller_id
- order_placed
- buying_mode
- buyer_organizationName
- buyer_departmentName
- consignee_id

Model Methodology:

The required data is extracted from database; The data is processed and categories such as custom and BOQ gets excluded, Buyer behavior and categorical behavioral is achieved from data processing. Similar to collusion there are two methods in which we can find out collusion. Initially using Machine learning module created to identify the suspected transactions. But later by discussing with GeM the rule based approach was defined and finalized to get outcome which provides better expected result.

1. Rule Based model (Suggested by GeM)
2. Machine learning model (Unsupervised model)

Steps In Model:

1. Extract data from database
2. Data pre-processing
3. Get Buyer behavior
4. Get Category Behavior
5. Get buyer-category behavior
6. Data processing and aggregation; and apply rule based approach.
7. Create final report

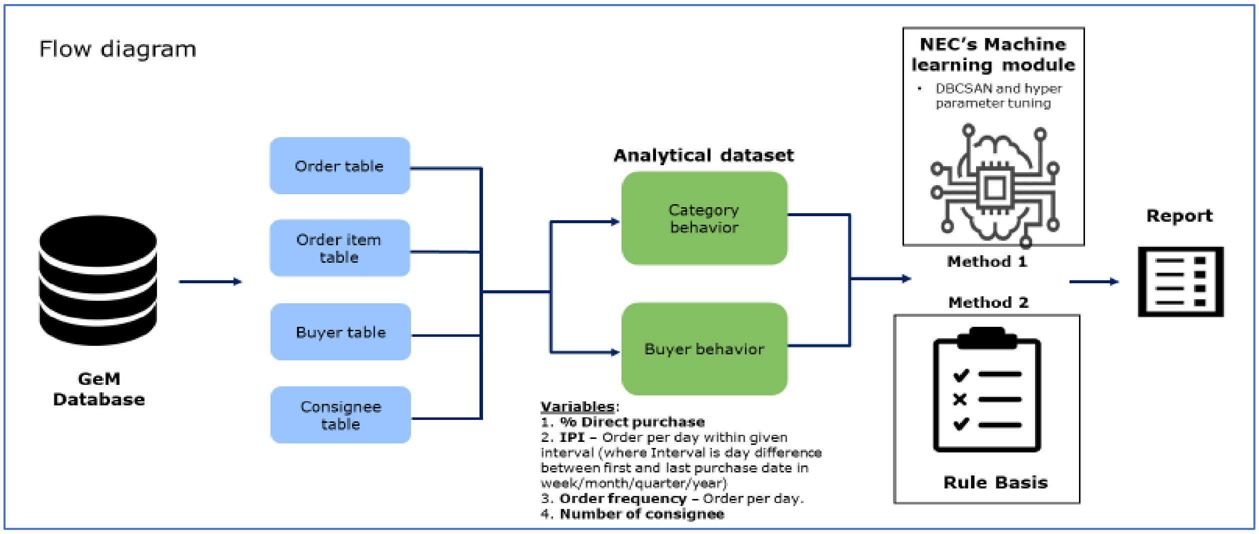


Image shows process diagram of avoiding order aggregation

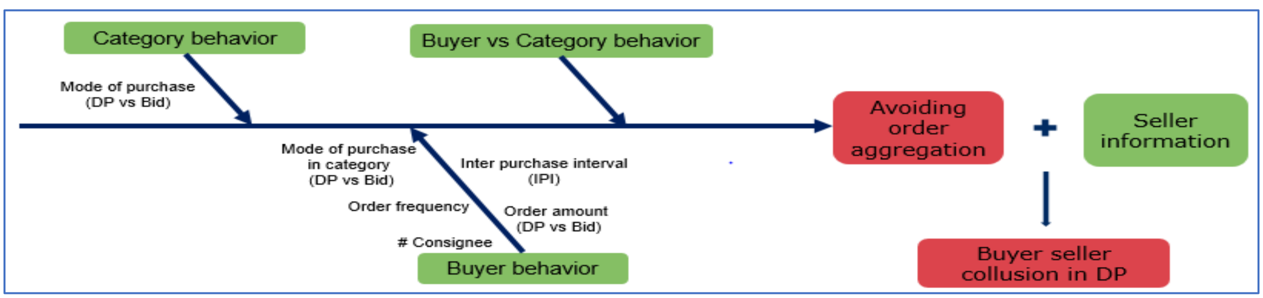


Image shows factors to define avoiding order aggregation

Business Value:

1. Improved Transparency and Efficiency in Public Procurement: The solution aligns with GeM's objective of enhancing transparency and efficiency in public procurement by identifying malicious activities.

2. Promotes Bid process: Solution helps buyer to create bids by prohibiting splitting of transactions through direct purchase. Bidding encourages suppliers to compete, which can lead to innovation and lower prices to ensure fairness on portal by allowing all interested parties to participate.

XII. Image Duplication:

Business Requirement:

GeM mandates every product to have at least 3 different images of a product, to help buyer understand what one is procuring / paying for. This helps to cover all physical aspect of the entire product.

However, majority of the sellers circumvent the system by uploading the same images over and over with minor modification. This undermines the purpose of the policy implemented.

A sanitized marketplace has a good representation of the product, and the product image is the most important component to do so.

To maintain the quality of the marketplace GeM wants to identify, prevent and remove the product catalogues with duplicate images.

- GeM intends to give an alert in future to the sellers at the time of uploading the images if they are duplicate.

- Gem wants to identify which product have duplicate images currently in system and remove duplicate images of the product.

Proposed solution:

The proposed solution is a system to automatically detect and prevent the upload of duplicate product images on the GeM platform. This system will:

- Analyse the images uploaded by sellers.
- Identify duplicate images.

The objective of the Image Duplication model is to identify the duplicated images from the seller's uploaded images based on:

- Using an uploaded image as a base and checking the rest uploaded images for duplication.

The solution will involve building a machine learning model capable of detecting duplicate images. This model will compare the images using difpy, a python library which is used for detecting duplicate images.

Solution Consumption:

Following are the stakeholders and processes where this solution can have impact:

- Stakeholders – There are three stakeholders in the process:
- Buyers: The buyer visits the marketplace to select the product of interest to add it to the cart and initiate the procurement process. The misrepresentation of the physical appearance of the product can misguide the buyer while selecting it.

- Sellers: The sellers upload images and manage their catalogues in CMS and the same is reflected in the marketplace. Due to sellers' s lack of awareness about GeM norms regarding image upload that is to upload 3 different images of the product from different angles such as front, back, top, side, back, interior, close-ups or to show off the technology or science behind the product; sellers upload duplicate images of product. Duplicate images might hamper the chances of creating a good presentation of the product and hence can lead to low sales.

- GEM SPV: Duplicate images can undermine market sanitization and the quality of the marketplace.

Marketplace Sanitization

Identification of catalogues with duplicate images

Duplicate Image Quality Gate

Restricting the seller from uploading duplicate images in CMS

Data:

Images used are the product images available on GeM website. The required data can be fetched from the tables below:

- Table: inb_variants
- Table: inb_images

Model Methodology:

The code provided aims to automate the process of fetching, processing, and analysing product images for potential duplicates. The methodology involves several steps to extract relevant data from a MySQL database, process image URLs, download images, detect duplicates, and store the results for further analysis. Below is the methodology:

1. Data Extraction:
   - The process starts with extracting relevant data from the gem_staging database. Specifically, it fetches product variants, their associated images, and related metadata based on specific category IDs.
   - A structured SQL query is executed to retrieve only active variants, ensuring the data is relevant and up to date.

2. Data Processing:
   - The extracted data is processed to correct image URLs based on specific patterns. This ensures that all images are fetched from the correct server locations.

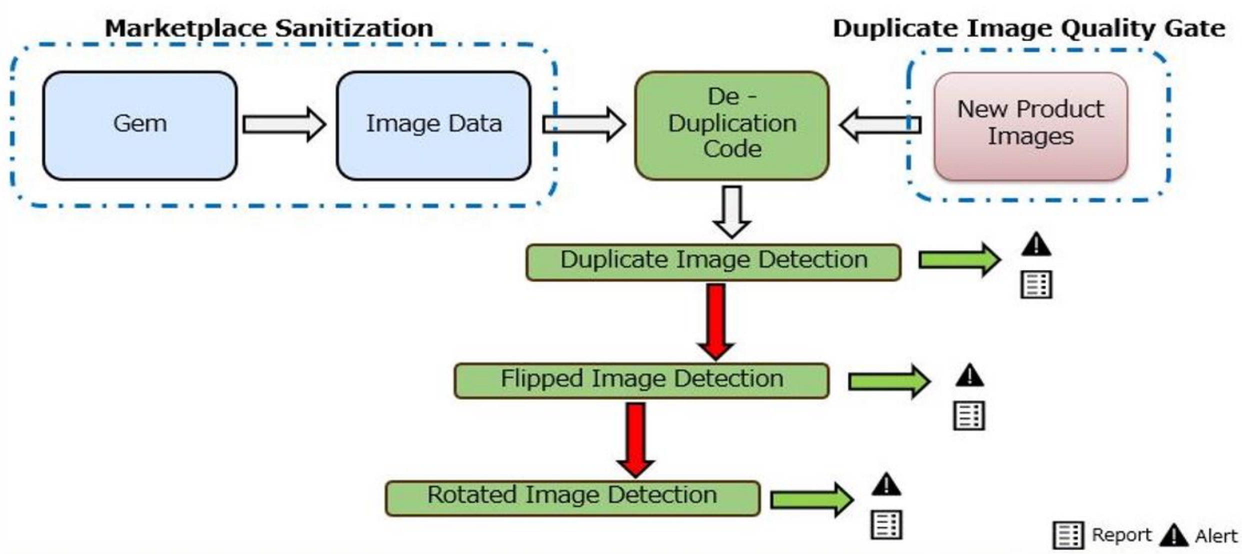
- Image URLs are adjusted to reflect the standard image format and path, which is necessary for consistent processing and storage.

3. Image Download and Storage:
   - For each product variant, all associated images are downloaded and saved in a uniquely generated folder. This ensures that images from different categories or variants do not overlap.
   - The folder naming convention eliminates any special characters that could cause issues in file path resolution.

4. Duplicate Image Detection:
   - The downloaded images are analysed for duplicates using the difPy library. This is done by comparing the images at a specified pixel size to identify those that are visually identical or similar.
   - The lower-quality duplicates are identified and listed, allowing for the retention of only the highest-quality images.

5. Data Storage and Reporting:
   - If duplicates are detected, the information is stored in a CSV file, detailing the duplicate images and their associated product URLs. This file is organized by category ID, making it easier to trace back to the original product data.
   - All images, whether duplicates or not, are moved to a final directory specific to each category. This allows for easy access and future reference.

See the picture below to know the flow of processes in this solution:



Steps In Model:

1. Extract Data from Database:
   - Execute SQL queries to extract product variant and image data for each category ID.
   - Store the extracted data in CSV files named after the respective category IDs.

2. Process Image URLs:
   - Filter the Data Frame for relevant image URLs.

- Correct the image URLs to point to the appropriate server paths.
- Ensure all URLs conform to a standard format for easier processing.

3. Generate and Manage Folders:
   - Generate unique folder names using UUIDs to prevent conflicts.
   - Create directories to store downloaded images for each category and variant.

4. Download Images:
   - Loop through each variant and download all associated images.
   - Save the images in the uniquely generated folder.

5. Detect Duplicate Images:
   - Use the difPy library to detect duplicate images in the folder.
   - Identify and list lower-quality duplicates for removal or analysis.

6. Store Duplicate Information:
   - Save the list of duplicates along with their corresponding product URLs to a CSV file.
   - Organize the CSV file by category for easy reference.

7. Move Images to Final Directory:
   - Transfer all downloaded images, including detected duplicates, to a category-specific folder for long-term storage.

This approach ensures that the process is automated, systematic, and organized, allowing for efficient management of large datasets involving product images.

Business Value:

1. Improved Marketplace Quality**:**
   By ensuring that all product images are distinct and accurately represent the product, GeM can maintain a high-quality marketplace, which is essential for buyer trust and satisfaction.

2. Enhanced Seller Accountability**:**
   Sellers will be encouraged to adhere to platform policies, knowing that the system will flag any attempt to circumvent the rules.

XIII. Image Category Relevance:

Business Requirement:

The GeM portal requires an automated solution to ensure that product images align with the correct category specifications. This is critical for maintaining accurate product categorization and improving the quality of listings on the platform.

Currently, the manual oversight process is inconsistent and prone to errors, which affects the reliability of the marketplace. The goal is to implement an AI/ML-driven module to automatically verify that product images meet the specified category features, thereby enhancing accuracy and reducing manual intervention.

Proposed Solution:

The solution involves developing a Category Relevance module using a Object detection model, specifically leveraging YOLOv8. This module will automate the identification and verification of features in product images to ensure that they correspond with the required category specifications.

By utilizing image annotation data, the system will be able to train a model capable of detecting these features in new images, ensuring accurate product categorization on the GeM portal.

Solution Consumption:

1. Market Sanitization:

This process will ensure the quality and accuracy of the marketplace by identifying and removing products with images that do not meet category specifications. The module will generate detailed reports for GEM SPV, highlighting any discrepancies between the image features and the category requirements.

2. Quality Gate (CMS):

A quality gate mechanism will be established to prevent sellers from uploading images that do not meet the required specifications. Alerts will be generated for GEM SPV and the sellers when an image fails to meet the criteria, ensuring compliance with category standards before the product is listed.

Data:

1. Image Data: Product images sourced from GeM marketplace. The required data can be fetched from below tables from the database.
   - inb_variants
   - inb_images

2. Annotation Data: Images are annotated to identify category-specific features, which are crucial for training the object detection model.

Model Methodology:

1. Image Annotation:
   - Product images are annotated to mark relevant features based on the specific category requirements.

2. Model Training:
   - The annotated data is used to train a YOLOv8 object detection model. This model is refined to accurately detect features that are relevant to the category specifications.

3. Model Evaluation:
   - The trained model is evaluated to assess its accuracy in detecting the annotated features.

4. Model Testing:
   - The model is tested on new images, where the detected features are compared against the Product Data to verify their relevance to the category.

Steps In Model:

1. Data Collection: Gather image data from GeM source.

2. Image Annotation: Annotate the images to highlight category-specific features.

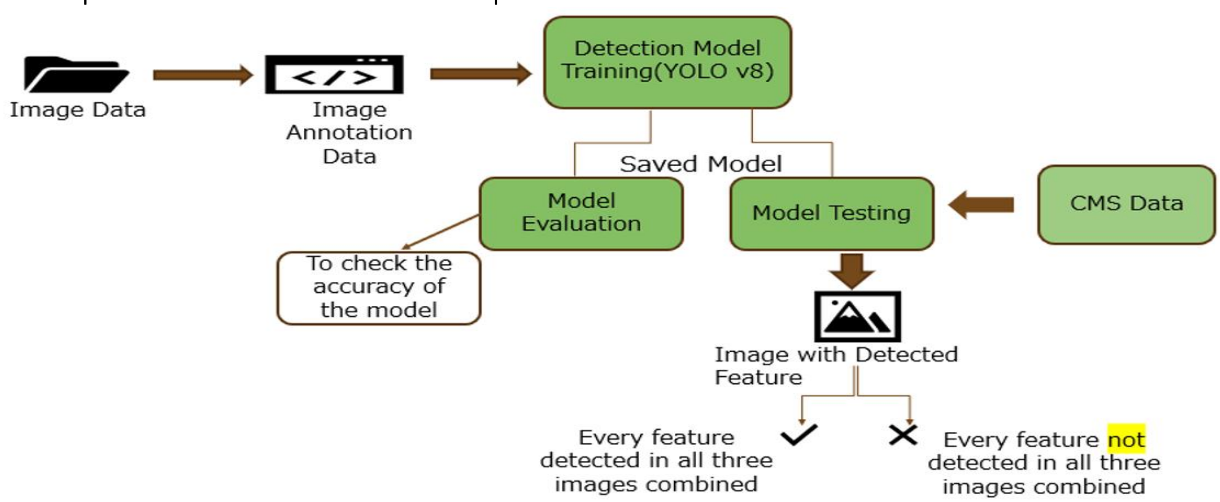
3. Model Training (YOLOv8): Train the YOLOv8 model using the annotated images.

4. Model Evaluation: Evaluate the model to ensure it correctly identifies the relevant features.

5. Model Testing: Test the model on new images and compare results with Product data to validate category relevance.

6. Deployment: Created a API to consume the functionality and deployed on the server.

See the picture below to know the flow of processes in this solution:



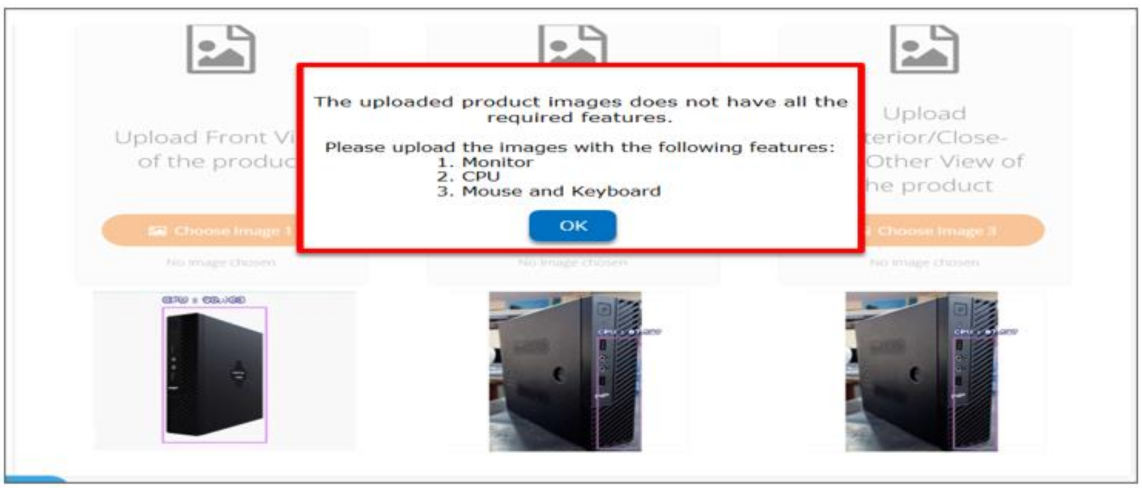
Business Value:

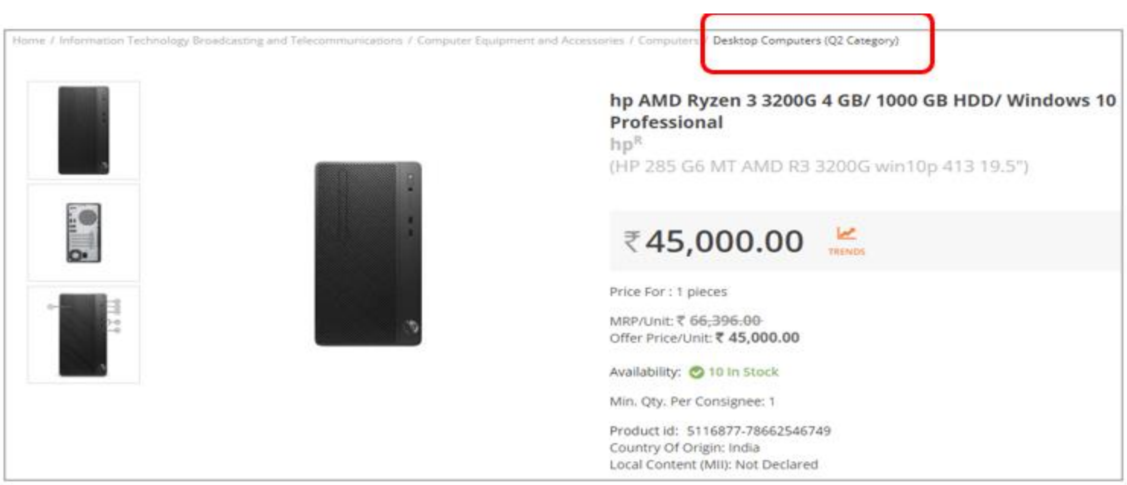
1. Enhanced Accuracy in Categorization: Automating the feature detection and categorization process improves the accuracy of product listings, ensuring they are correctly categorized based on their images. This reduces the potential for manual errors and oversight.

2. Improved Marketplace Quality: By ensuring that only accurately categorized products are listed, the solution enhances the overall quality and reliability of the GeM platform, fostering better user trust and satisfaction.

3. Efficient Quality Control: The quality gate mechanism prevents non-compliant images from being uploaded, streamlining the process of maintaining high standards for product listings. This proactive approach minimizes the need for post-upload corrections, improving operational efficiency.

Solution To-be:





XIV. Image Category Mismatch:

Business Requirement:

The Category Mismatch module for the GeM portal addresses issues of products being incorrectly classified into categories, either due to intentional miscategorization or unintentional errors by sellers. The goal is to identify products whose images do not align with their assigned categories, thus enhancing buyer trust and improving the quality of product listings on the portal. The model analyzes product images and classifies whether they fit within the selected category, flagging any anomalies or potential manipulations.

Proposed Solution:

The solution involves using a CNN model to categorize products based on images. The trained model is capable of classifying images into the correct categories and flagging products that do not match their designated category as anomalies.

Solution Consumption:

1. Market Sanitization:- The module contributes to the Market Sanitization process by continuously monitoring and analyzing product images across the marketplace. It identifies and highlights products that do not meet the specified category alignment. Detailed reports will be generated for GeM SPV, showcasing discrepancies between product images and their respective categories. This allows for timely intervention and maintenance of marketplace integrity.

2. Quality Gate(CMS):- The Category Mismatch module is currently being used in the Catalog Management System (CMS). It automatically checks whether the product images match their assigned categories. When the module finds mismatches, it flags these products for review. This helps to ensure that products are listed correctly, reducing errors and preventing sellers from intentionally placing items in the wrong categories. This keeps the platform accurate and easier to use for buyers.

Data:

The required image data can be fetched from below tables to detect the anomalies present in the categories in GeM marketplace:
- inb_variants
- inb_images

Model Methodology:

The Category Mismatch Detection Module utilizes deep learning to identify products whose images don't match their assigned categories. The methodology involves:

1. Data Collection:
   Uses product images from the GeM portal and external e-marketplaces, organized by category.

2. Data Preprocessing:
   Images are resized to 224x224 pixels, with data augmentation applied to ensure a balanced dataset across all categories.

3. Model Training:
   A CNN model is trained for multi-class classification. It uses convolutional layers, pooling, dropout, and dense layers. The Adam optimizer is employed, and models are trained for 6 epochs.

4. Model Dictionary:
   A dictionary links categories to their trained model paths, allowing the correct model to load for evaluation.

5. Evaluation:
   For each product image, the relevant model is loaded, and predictions are made. If the confidence score is below 0.08, the product is flagged as an anomaly.

6. Report Generation:
   Results are saved as reports, identifying images that potentially mismatch their assigned category.

Steps In Model:

1. Data Collection:
   Gather images from GeM database.

2. Data Preprocessing:
   Resize images and apply augmentation for balance.

3. Model Training:
   Train CNN models for category-based image classification.

4. Prediction:
   Load the relevant model, predict the category, and flag anomalies based on confidence scores.

5. Report Generation:
   Create reports detailing potential mismatches.

Business Value:

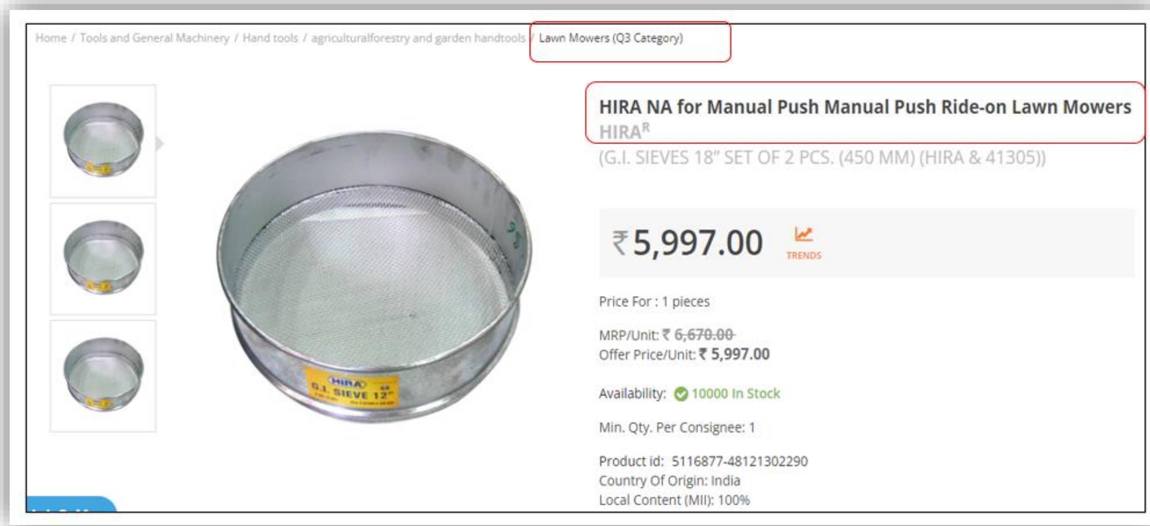
1. Improving Buyer Experience:
   Ensuring that buyers see accurately categorized products increases trust and satisfaction.

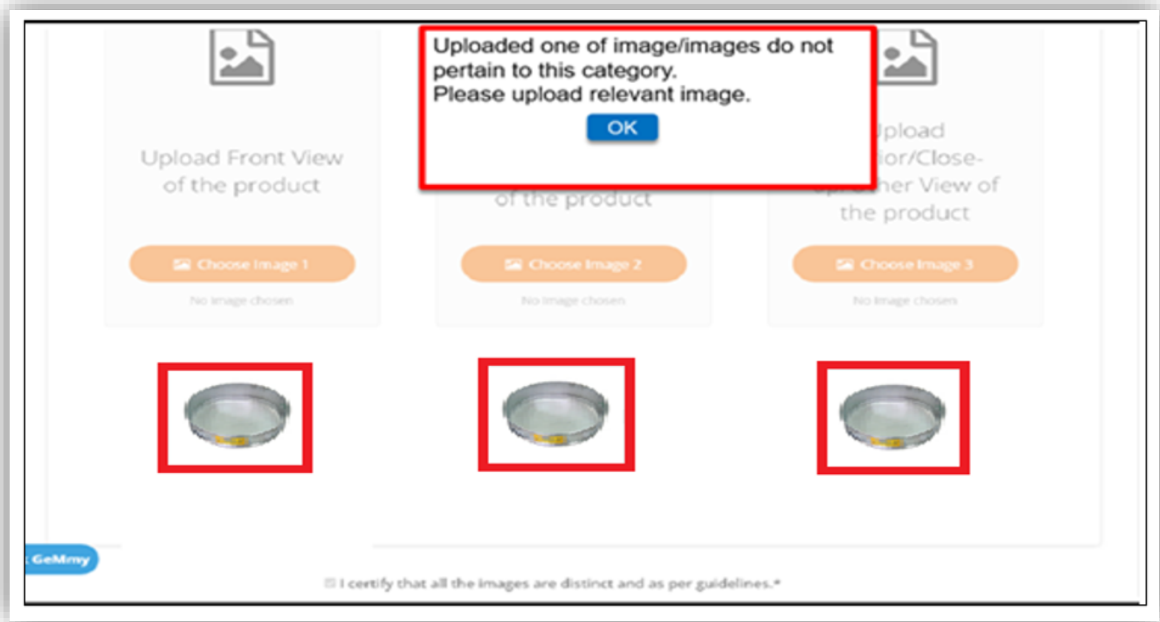
2. Automation of Category Validation:
   The module reduces manual intervention, speeding up the product approval process.

3. Fraud Prevention:
   By detecting intentional miscategorization, the module helps in preventing fraudulent listings.

4. Compliance with Standards:
   Ensures that product categories adhere to the guidelines set by the GeM portal, enhancing overall platform reliability.

Solution To-be:





XV. Certificate Validation:

Business Requirement:

In GeM portal, seller provides different certificates such as MRP documents, registered trademark certificate, DPIIT certificate and many other documents. Buyers also need to provide some documents such as CA (Competent Authority) document approval. Currently validation of the information provided in these documents is a manual process.

GeM intends to automate/semi-automate validation of the information provided by sellers/buyers in different certificates with the information already provided by same sellers / buyers during on-boarding/bidding process. This is with intention to save time in verifying documents and avoid errors caused by manually verification. This will help with preliminary check on the documents uploaded, helping sellers/buyers avoid any mistaken upload of documents by the stakeholders.

GeM requires a scalable solution for validating products and services related documents. Data extraction for analytics consumption from unstructured data/ scanned documents submitted. Content mining, Entity matching and Named Entity Extraction for capturing key parameters from the scanned documents which are not covered as part of overall QCI scope to identify patterns and generate actionable insights to design/improve future processes like vendor on-boarding, bidding etc.

Proposed solution:

Develop a data extraction model from scanned images/pdfs and a validation mechanism to verify information provided in certificates. Certificate's validation process should be automated or semi-automated. This would reduce the time it takes to validate the documents and eliminate errors caused by manually importing the data into the report. As already mentioned in above section, which are the processes in which document verification performed in GeM. Once this solution applied, information obtained will be used for taking decisions

Generic system to validate the content within the document based on the information provided by the sellers / buyer:

- Certificate conversion from PDF to image
- Image enhancement for increasing readability
- Data extraction from the enhanced image
- Data processing and NLP (Fuzzy Wuzzy)
- Data validation as per the user details in Database.

Solution Consumption:

Following are the stakeholders and processes where this solution can have impact. There are two stakeholders in the process:

- Buyers: Need to manually validate information provided by sellers in bidding process like MRP certificate, Trademark certificate etc.

- GEM - MSP: The team manually validates the documents provided to see if the information in the submitted documents by the buyer / seller matches the database information.

Data:

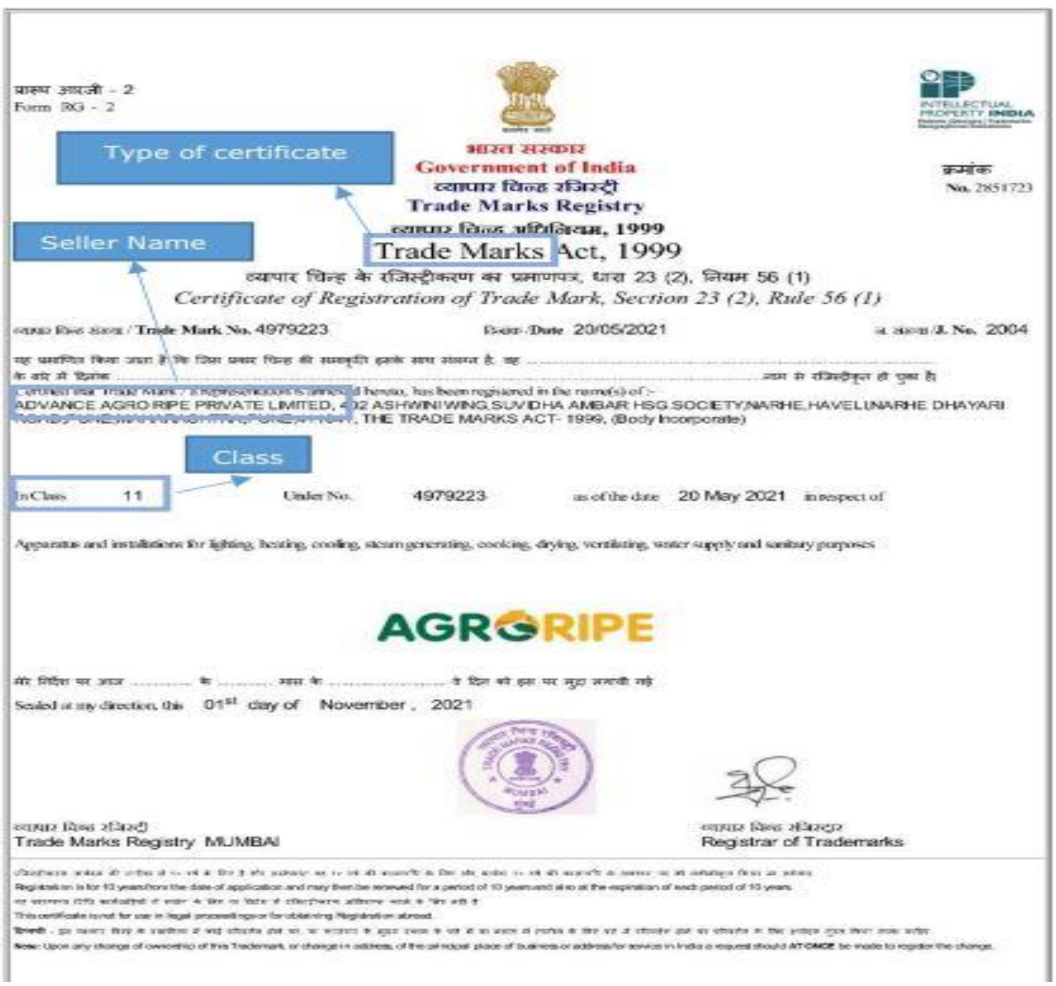
- We require 7 documents (DPIIT, Trademark, MRP, FSSAI, Meity, PSARA, BIS) which are uploaded by Seller / buyer image or pdf format.
- We require the concerned seller details of whom the documents have been uploaded.
- We require the concerned Buyer details of whom the documents have been uploaded.

How to get the required data:

- Documents Data shared by GeM-MSP team for all the above-mentioned types of documents with all variations to create solution. At the time of ‘go-live’ on production system, GeM-MSP team to share document to be validated at process of GeM.
- NEC can connect to the database and pull the required details to perform validation.

Eg: Documents Data:

Below is the example of Trade-mark certificate and the information that need to be extracted. The information from certificates extracted and validated with database information using NLP techniques.



Form RG - 2

भारत सरकार
Government of India
Trade Marks Registry
Trade Marks Act, 1999

No. 2851723

व्यापार चिन्ह के रजिस्ट्रीकरण का प्रमाणपत्र, धारा 23 (2), नियम 56 (1)
Certificate of Registration of Trade Mark, Section 23 (2), Rule 56 (1)

Trade Mark No. 4979223 Date 20/05/2021 J. No. 2004

has been registered in the name(s) of :-
ADVANCE AGRO RIPE PRIVATE LIMITED, 402 ASHWINI WING,SUVIDHA AMBAR HSG SOCIETY,NARHE,HAVELI,NARHE DHAYARI ..., THE TRADE MARKS ACT- 1999, (Body Incorporate)

In Class 11 Under No. 4979223 as of the date 20 May 2021 in respect of

Apparatus and installations for lighting, heating, cooling, steam generating, cooking, drying, ventilating, water supply and sanitary purposes

AGRORIPE

Sealed at my direction, this 01st day of November , 2021

Trade Marks Registry MUMBAI

Registrar of Trademarks

Registration is for 10 years from the date of application and may then be renewed for a period of 10 years and also at the expiration of each period of 10 years.

This certificate is not for use in legal proceedings or for obtaining Registration abroad.

Note: Upon any change of ownership of this Trademark, or change in address, of the principal place of business or address for service in India a request should AT ONCE be made to register the change.

Eg: Database Data:
Data available in database, will be used for validating certificate information.

Seller Name	Class
Advance agro ripe private limited	11

Table: Required information in database

Model Methodology:

The code is designed to semi-automate the validation of information provided in various certificates by sellers and buyers on the GeM portal. The methodology involves the following steps:

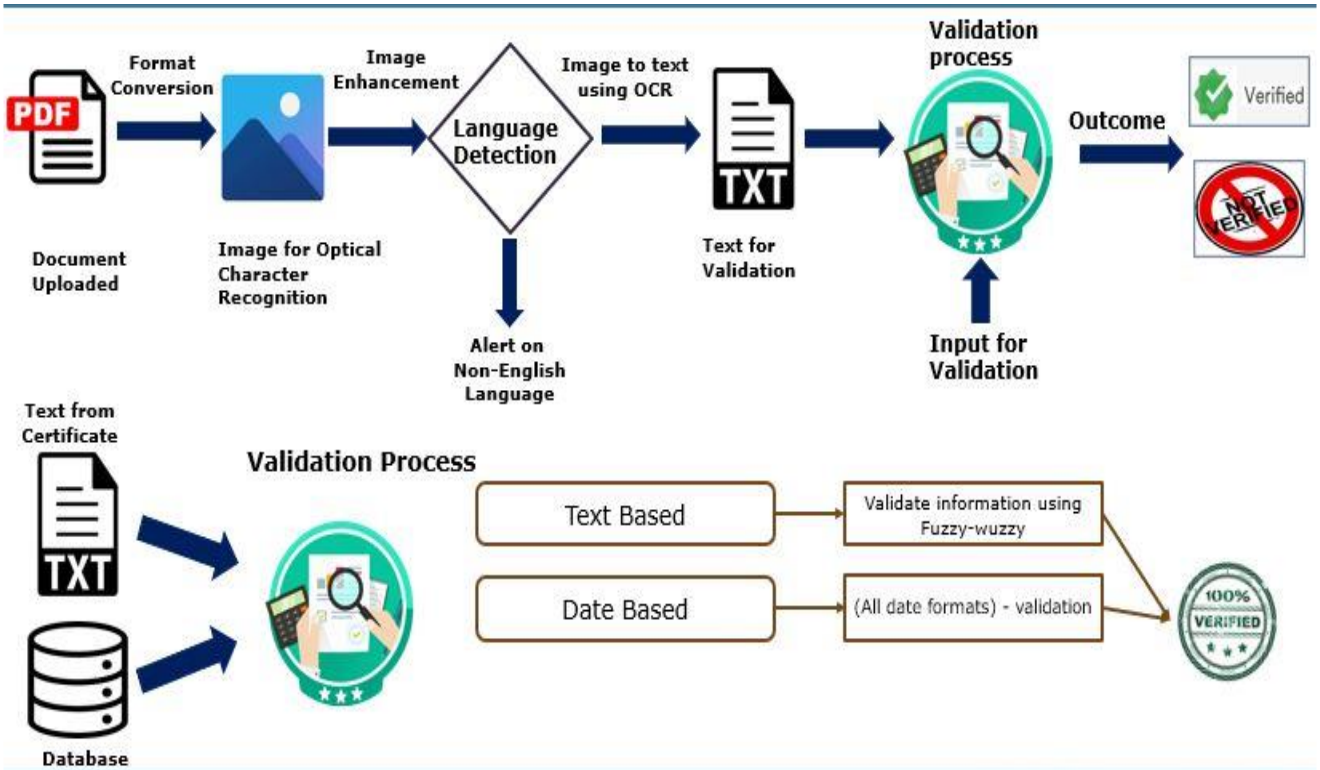
1. Text Data Processing**:**
   - The code begins by extracting and processing text data from a provided document (doc). This document might be extracted from certificates submitted by users.
   - The text is cleaned by converting it to lowercase, removing newline characters, and stripping any unnecessary whitespace. This step ensures the text is in a uniform format for further processing.

2. Fuzzy Matching Using NLP Techniques**:**
   - The core function (test1) takes a search string (which could be critical keywords or phrases from the on-boarding/bidding process) and splits it into individual words.
   - The document is also split into words, and each word in the search string is compared against different parts of the document using fuzzy matching techniques.
   - The fuzzy matching is done using the fuzzy Wuzzy library's token_set_ratio function, which calculates similarity scores between the words in the search string and sections of the document.
   - The function is versatile and can handle search strings of varying lengths (from 1 to 7 words), ensuring that different document types and content lengths are appropriately handled.

3. Data Storage and Analysis:
   - The similarity scores calculated during the fuzzy matching process are stored in a Pandas Data Frame, with columns corresponding to each word's matching score.
   - This structured data allows for easy analysis, where matching scores can be reviewed to determine the accuracy of the document content against the provided information during the on-boarding/bidding process.

4. Validation Mechanism**:**
   - The resulting scores provide a basis for validating the information contained within the documents. By comparing the scores against a predefined threshold, the system can determine whether the document content aligns with the expected information.
   - This semi-automated approach helps in the preliminary check of the documents, reducing the likelihood of errors and speeding up the validation process.

See the picture below to know the flow of processes in this solution:



Steps In Model:

1. Extract and Preprocess Text Data:
    - Input Document: Start by loading the document (doc) that contains text extracted from certificates or other relevant sources provided by sellers or buyers.
    - Text Cleaning: Clean the document text by converting it to lowercase, removing newline characters, and stripping any unnecessary whitespace. This standardizes the text for consistent processing.

2. Search String Preparation:
    - Search String Input: Define the search string, which represents the critical information that needs to be validated against the document. This could be key phrases, entity names, or other relevant data from the on-boarding/bidding process.
    - Word Splitting: Split the search string into individual words, preparing them for detailed comparison against the document content.

3. Fuzzy Matching of Text:
    - Document Splitting: Split the document into a list of words, preparing it for matching.
    - Token Set Ratio Calculation: For each word in the search string, calculate the similarity score between the word and various sections of the document using the fuzzy Wuzzy library's token_set_ratio function.

- Handle Different Search String Lengths: Adapt the matching process to handle search strings of varying lengths (from 1 to 7 words), ensuring flexibility in dealing with different types of documents.

4. Store and Analyse Matching Scores:
   - Data Frame Creation: Store the calculated similarity scores in a Pandas Data Frame. Each word's score is saved in a separate column, providing a structured view of how well the document matches the search string.
   - Score Review: Analyse the scores to determine the accuracy of the document's content against the expected information provided during the on-boarding/bidding process.

5. Validation Mechanism:
   - Threshold Comparison: Compare the similarity scores against predefined thresholds to assess whether the document content aligns with the expected data.
   - Decision Making: Use the comparison results to make decisions on the validity of the document. Documents that do not meet the threshold criteria may require further manual review or be flagged for errors.

Note:

As per our understanding and after discussion with GeM taskforce, below are the processes and documents identified in which this verification solution can be applied in GeM.

- *Seller On Boarding*: DPIIT, DOE order compliance (At the time of profile update) and Seller profile update documents.
- *CMS*: Trademark certificate, MRP, Unbranded unregistered etc. certificates
- *Bidding related*: Buyer side Competent Authority document

Business Value:

1. Reduces the time it takes to validate a record by ten times.

2. Verifies that the buyer/seller has uploaded the correct certificate.

3. Tries to capture the certificates where validity has been expired or about to expire

4. GeM stakeholders can identify Buyers/Sellers in case they are uploading fake certificates / Irrelevant certificates.

XVI. Health Of Bid

Business Requirement:

Today, bidding process is seen as an enabler for fair competition. However, there is no mechanism to understand the effectiveness of the process. In other words, there is no measure of health of a bidding process. To aware buyer where their bid falls on basis of participation and price.

Currently, there is no established process to retrieve such intelligence from the ongoing bidding process. The buyer lacks the necessary intelligence to assess the effectiveness of the bid created. Similarly, sellers do not have visibility into how their bids are compared to competitors' offerings, leading to potential misunderstandings, and missed opportunities.

Analysis of overall effectiveness of the bidding processes across product categories and buyers & utilize historical bidding data to identify trends and patterns requiring any policy intervention due to gaps in current processes.

- Are buyers getting any benefits in bidding process as compared to direct mode?
- Determine trend of impact on product category prices with increased number of suppliers to ensure benefit being passed on to the buyers resulting in buyer acquisition.

Concept: Analyse the Current bid with its past similar bids on basis of four parameters and provide measure to its performance.

- To check where Current Quoted L-1 Price falls under,
- To check the Price variation from similar bids
- To check for participation
- To check for qualified sellers

Proposed solution:

To enhance buyer decision-making and bid optimization on the GeM portal, proposed solution leverages data from similar bids over the past six months and by using Product Similarity module. This involves extracting key information such as Participation (number of bidders), Price, and Qualified Sellers from these historical bids. Statistical significance tests will then be employed to determine price ranges and assess where the current L1 Price, Participation, and number of Qualified Sellers fall within these ranges.

Based on the identified price ranges and statistical analyses, buyers will gain insights into how their current Key Performance Indicators (KPIs) — such as L1 Price, Qualified prices, Participation, and Qualified Sellers — compared to historical data. This information will empower buyers to make more informed decisions, including the ability to negotiate with sellers more effectively to ensure competitive pricing and optimal bid participation.

The proposed solution will analyze historical bid data to provide insights into the "health" of a bid. By leveraging past bidding patterns and outcomes, the solution will offer a comprehensive view of bid effectiveness. This will help buyers understand how well their bids align with successful bidding practices and give sellers a better perspective on their competitive positioning.
This enhanced visibility into bid health based on historical data will improve decision-making for both buyers and sellers, leading to more efficient and successful bidding processes.

Solution Consumption:

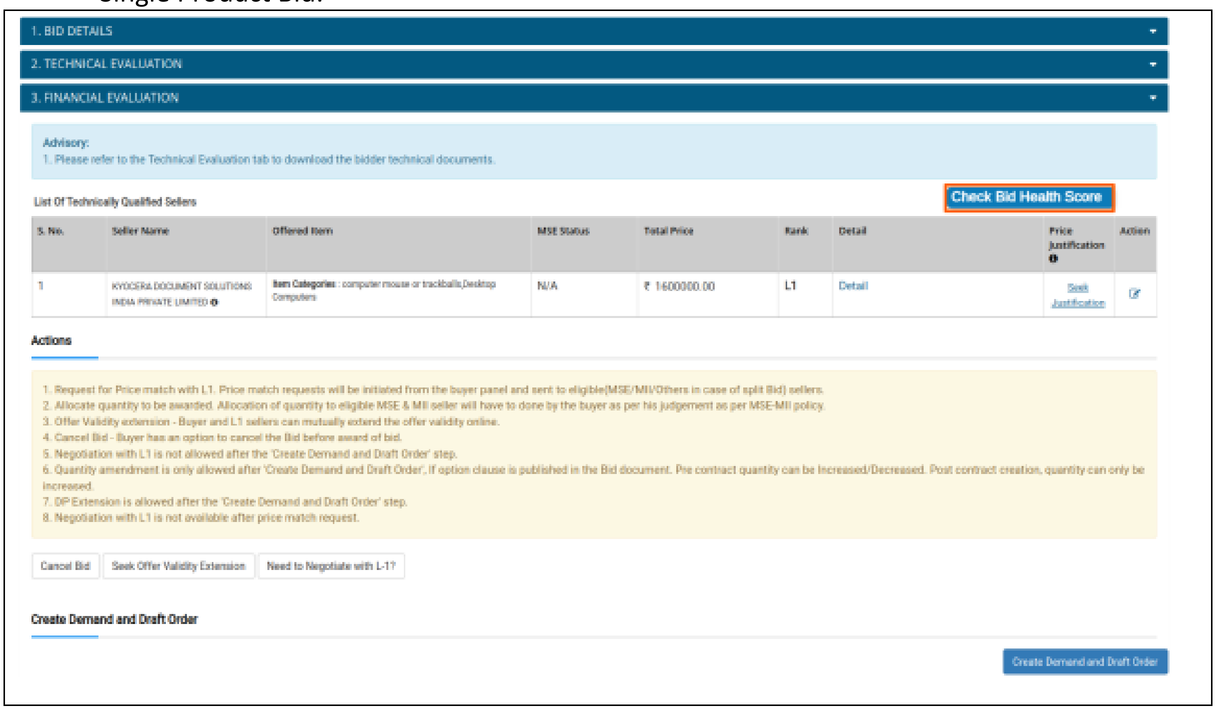
The following are the stakeholders and processes impacted by this solution:

- Buyer: The primary stakeholder who will benefit from this solution by gaining insights into the effectiveness of their bids.
- Seller: Sellers will also benefit as the solution provides a clearer understanding of how their products are evaluated against others, reducing discrepancies in bid rejections and improving their chances of successful bidding.

Currently, this solution is in UAT phase and then will be integrated GeM portal, at bidding process.

When buyer click on "Check Bid health score" tab, NEC API call will provide the data and data will be shown in tabular, if API result is success from NEC. Below are the samples of the page how it will be shown at stages of BID.
Note :- Check "Bid Health Score" button will be available to buyer after Bid award as well.

- Single Product Bid:



- Bunch Item Bid:

For bunch item bid, check bid health button will be available for each schedule on basis of evaluation method selected by buyer.

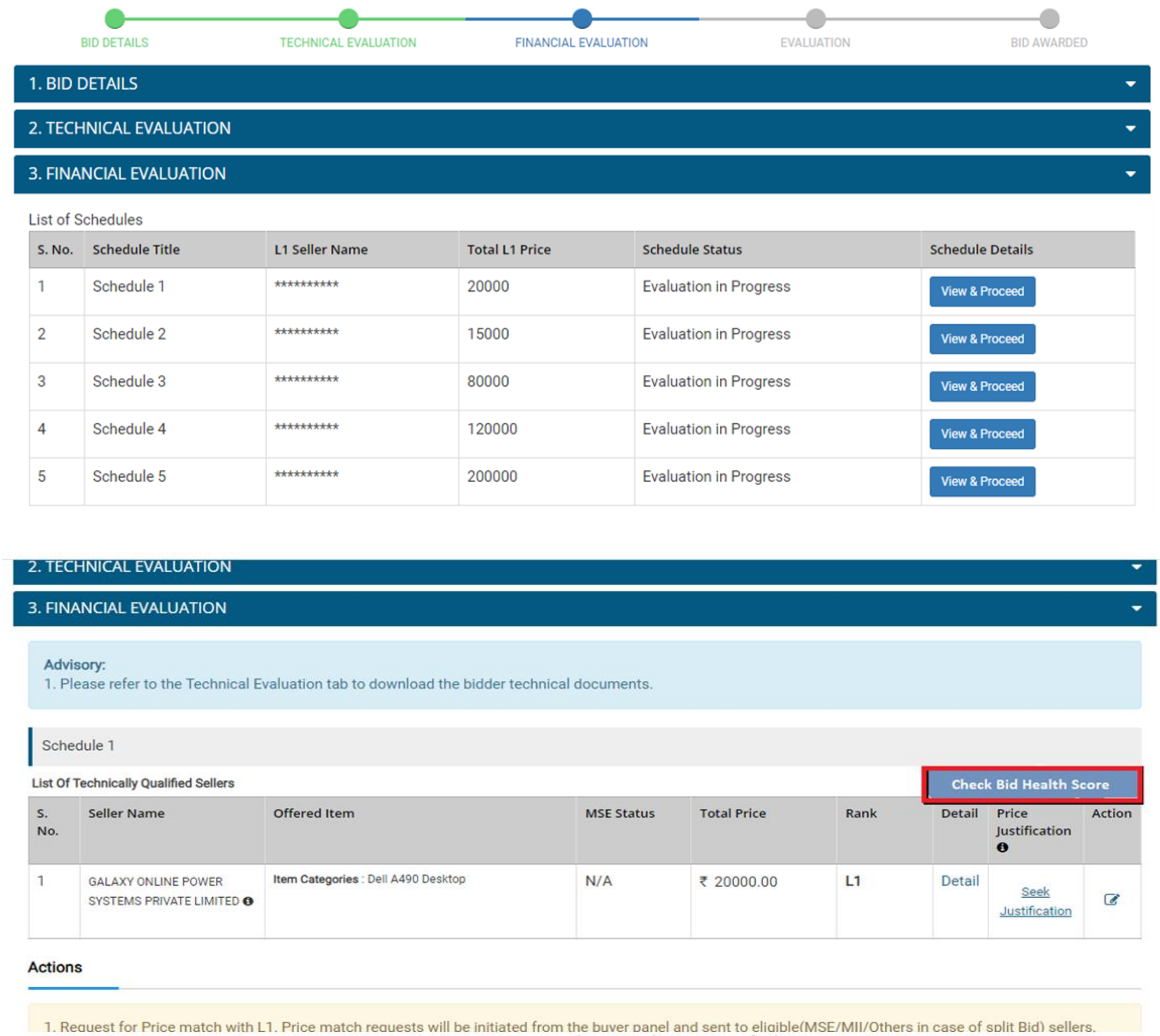


Image: Proposed UI on implementation of solution in GeM

Note: The UI format will be finalized and updated based on GeM Authority suggestions as per requirement

Final Bid Health score will be shown once buyer click on "Check Bid Health Score" as shown in below tables:

1. Single Product Bids:

L1 Price Score	Price Variance Score	Bid Total Participation Score	Bid Qualified Participation Score	Overall Bid Health Score
0.7	0.8	0.25	0.4	53

Min score	Max score	Color
81	100	Green (Good Health)
51	80	Orange (Medium Health)
0	50	Red (Low Health)

2. Bunch Product / Multi consignee Bids:

Multilevel L1 Price Score	Multilevel Price Variance Score	Multilevel Bid Total Participation Score	Multilevel Bid Qualified Participation Score	Overall Bid Health Score
0.9	0.7	0.8	0.6	65

Min score	Max score	Color
51	100	Green (Good Health)
26	50	Orange (Medium Health)
0	25	Red (Low Health)

**Note: The calculation format will be finalized and adjusted based on the color-coding requirements specified by the GeM Authority.*

3. For Packet bids, the qualification score will be ignored. The evaluation will focus on the other three parameters.
   Note: If sufficient data for similar bids is not available, the user will receive the following message:
   Message Advisory: "Bid Health Score not available since sufficient data of similar past bids are not available."

4. For Bunch Bids: The score will be calculated for each product and bid level, including Price_Score, Variance_Score, Participation_Score, and Qualification_Score along with the overall_score. Only the weighted average of the bid health score will be shared by the NEC team.

5. For Multi-Consignee Bids: Scores will be calculated at each product ID and consignee ID level.

6. Bid Score Data: All bid score data will be stored in logs (bid-specific) maintained by MSP (TCS).

7. Price Variance Score:
   - For Bid: Based on qualified bidders' prices in the bid.
   - For Bid to RA (Reverse Auction), the Price Variance Score is calculated as follows:
     - If a qualified bidder did not participate in the Reverse Auction (RA) or is not eligible for RA, then their price in the original bid is shared in bid and used to calculate the Price Variance Score.
     - If the bidder participated in the RA, then the RA price (the final price after the auction) is shared in bid and used for the Price Variance Score.

Data:

To build the bid health analytics solution, the necessary data can be retrieved from the following tables in the GeM database, which include bid details, participation information, and product information.

- bdp_odb_bids
- bdp_odb_bid_participation
- bdp_odb_bid_participate_details
- ffm_order
- ffm_orderitem
- bdp_bid_item_consignee
- bdp_bid_consignees
- bdp_odb_bid_details
- bdp_bid_attributes

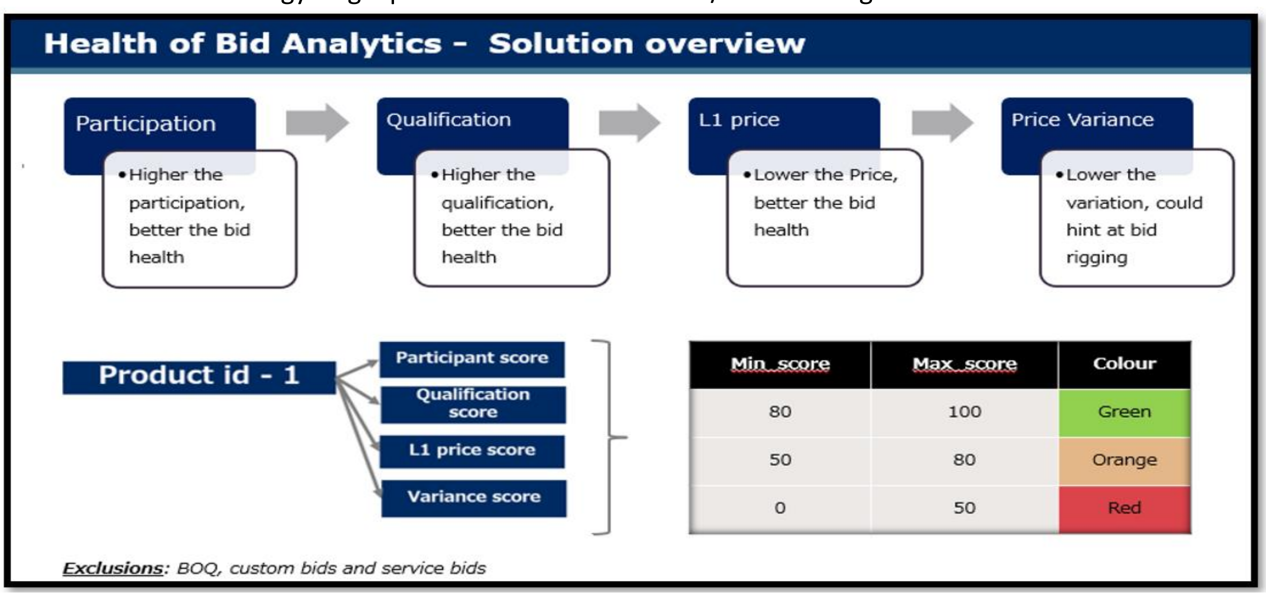
Model Methodology:

The model evaluates bids by leveraging 6 months of historical data and a Product Similarity module to assess the relevance of current bids. It calculates individual scores based on product attributes, price variance, and seller participation. For single product bids, it provides a direct score, while for bunch and multi-consignee bids, it consolidates scores across multiple products or consignees. The final output integrates these scores to offer a comprehensive assessment of bid performance against historical benchmarks.

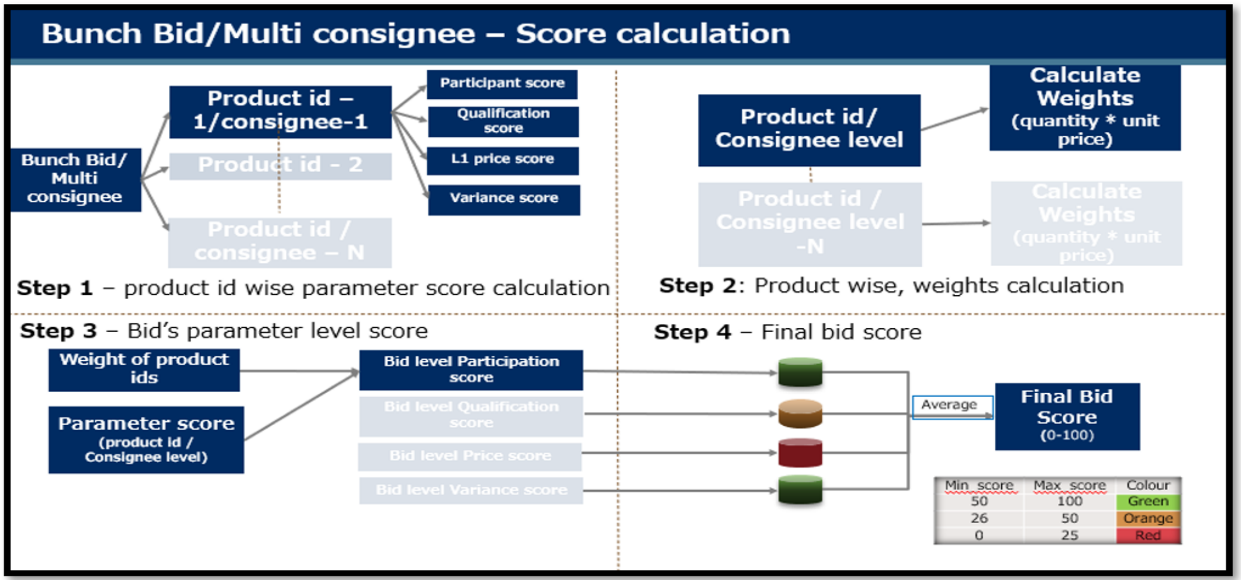
1. Parameters for Similar Bids:
   - Category ID: Bids are compared within the same product category.
   - Quantity:
     - For quantities up to 20 units: Bids within ±5 units.
     - For quantities above 20 units: Bids within ±25% of the required quantity.
   - Consignee Location: Bids are compared at the state level.
   - Product ID: Product Similarity score is used to identify similar products.

2. Matrices for Bid Performance:
   - L-1 Price of Current Bid: The lowest price offered in the current bid.
   - Variance of Price Quoted by Qualified Sellers: Measures the spread in prices among qualified sellers, indicating price competitiveness.
   - Participation in the Bid:
     - Qualified Participants: Number of qualified sellers who participated in the bid.
     - Total Participants: Total number of participants in the bid.

Below is the methodology single product bid and bunch bid / multi consignee bids:



Single Product Bids



Multilevel Bids

**Note: Custom Bids, BOQ and service bids are out of scope as solution has designed using concept of product's similarity, which is defined for Products.*

Steps In Model:

1. Import Libraries: Import necessary libraries and modules for data handling, analysis, and visualization.

2. Data Preparation:
    - Historical Data: Extract the required data, clean data (pre-processing) and utilize 6 months of historical data to provide a comprehensive view of bid trends and performance.
    - Product Similarity Module: Leverage this module to assess similarity bid scores of products, to get relevant similar historical products.

3. Data Normalization:
    - Convert JSON input to a normalized format suitable for analysis.
    - Structure data for compatibility with the Product Similarity module.

4. Bid Classification:
    - Single Bids: For single product bids, calculate a single product score.
    - Bunch Bids: For bids involving multiple products, calculate a multi-product score.
    - Multi-Consignee Bids: For bids involving multiple consignees, calculate a multi-consignee score.

5. Historical Data Matching:
    - Identify Historical Bids: Find historical bids that match the current bid's attributes (category ID, quantity required, consignee location, product ID).
    - Calculate Historical Similarity Scores: Use the Product Similarity module to determine similarity scores between historical bids and the current bid based on matched attributes.
6. Scoring Based on Historical Similarity:
    - Determine Historical Similarity: Score the current bid based on how closely it matches with historically similar bids. This includes comparing product attributes, price, and participation.

- Product Similarity Score: Integrate the Product Similarity score into the historical similarity score to enhance the evaluation.

7. Bid Comparison:
   - L-1 Price of Current Bid: Benchmark against the lowest price in the current bid.
   - Variance of Price Quoted: Calculate the variance in prices quoted by qualified sellers to assess competitiveness.
   - Participation:
     - Qualified Participants: Number of qualified sellers participating.
     - Total Participants: Total number of participants in the bid.

8. Bid Types and Specific Calculations:
   - Packet Bids:
     - Ignore qualification scores.
     - Use remaining three parameters: L-1 Price, Price Variance, and Participation.
   - Schedule Bids:
     - Calculate scores based on the schedule flag provided in the bids.
   - RA Bids:
     - Utilize historical data specific to RA bids based on the flag provided in the inputs.

9. Statistical Analysis:
   - Interquartile Range (IQR): Use IQR to identify and exclude outliers from price variance calculations.
   - Z-Score: Apply z-score to standardize data points and identify deviations from the mean, enhancing the comparison of bid prices.
   - Weighted Averages: Calculate weighted averages of prices and scores to give more importance to bids from more relevant or recent data(only for Bunch/Multi consignee bids).

10. Final Output: Based on Bid types and Flag indicators.
    - Single Product Bids: A single, comprehensive score for the single product bid.
    - Bunch Bids: A consolidated score for the entire bunch, reflecting the combined performance of all included products.
    - Multi-Consignee Bids: A consolidated score for the bid across all consignees, reflecting the overall performance.

Business Value:

1. Enhanced Buyer Confidence: Buyers have a clear, data-driven view of bid performance, enabling more informed and confident decision-making in the procurement process.

2. Improved Bid Selection: The health of bids identifies the most relevant and competitive offers, ensuring that procurement decisions are closely aligned with specific requirements and objective.

3. Optimized Procurement: For GeM, this approach streamlines the procurement process, ensuring better alignment between buyer needs and supplier offers, and ultimately enhancing the efficiency and transparency of the bidding process.

XVII. Technical Rejection Reasons (Comment Appropriateness)

Business Requirement:

At GeM portal in bidding while rejecting sellers, buyers must select a drop- down list of reasons for rejecting sellers and relevant comments must be provided for each reason.

Currently, some sellers are getting rejected by buyer without proper reasons & comments. In the present scenario, there’s no such module to find out the inappropriate remarks provided by buyer. GeM doesn't have a mechanism to validate the reasons specified by buyers and hence no way to check for unfair rejections.

By this solution, will get the list of the inappropriate comments and GeM management will take required actions on these scenarios for proper bidding process.

Proposed solution:

Develop Technical Rejection module with AI/ML algorithms/ business rules. This module will showcase irrelevant comments that are provide by buyer in rejections.
Note: This use case is not the part of the RFP and as suggested by CEO sir NEC developed this solution.

Solution Consumption:

Following are the stakeholders and processes where this solution can have impact.

- Buyers:
  The solution provides buyers with enhanced awareness through an automated comparison system that evaluates all offered products against a base product's technical specifications. While buyers retain the ability to select any product, the solution will present them with a clear comparison of each product's specification compliance. A "specification compliance score" will be generated for each product, which allows buyers to see how closely each offered product aligns with the base product's requirements. This transparency ensures that buyers are informed and can make more objective and data-driven decisions.

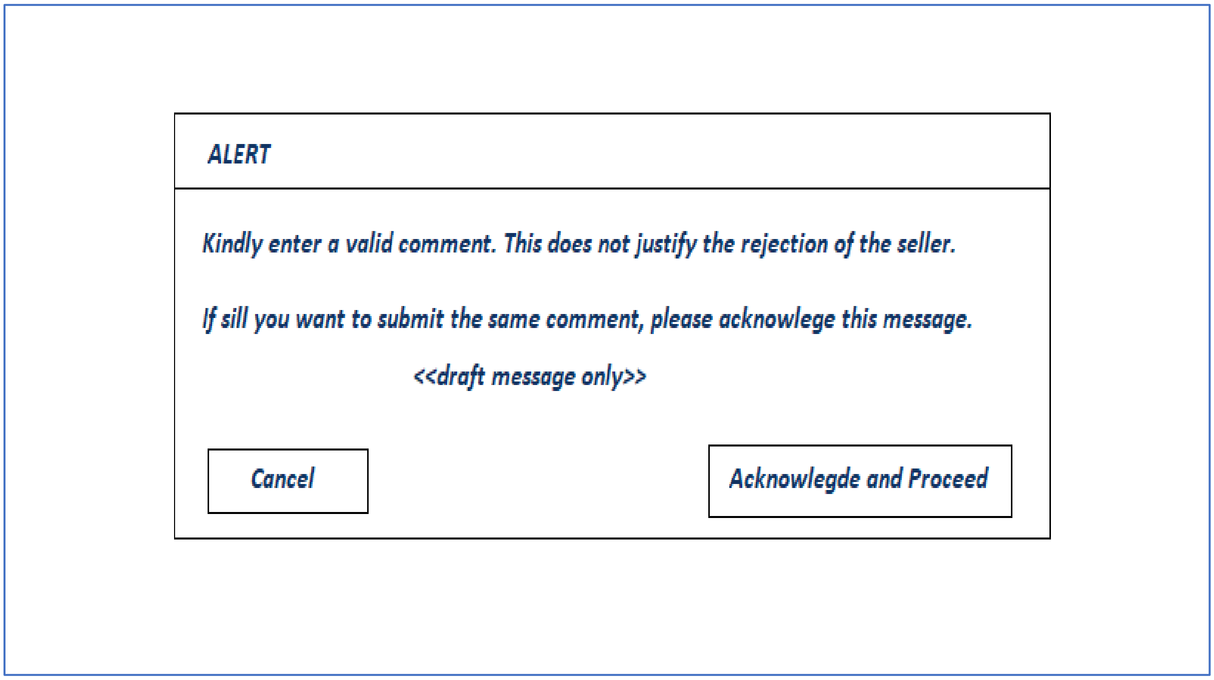
- Sellers:
  Sellers will benefit from increased transparency in the evaluation process. Products that meet the required specifications will be clearly identified, reducing the likelihood of rejection due to misinterpretation or oversight by buyers.

Consumption:

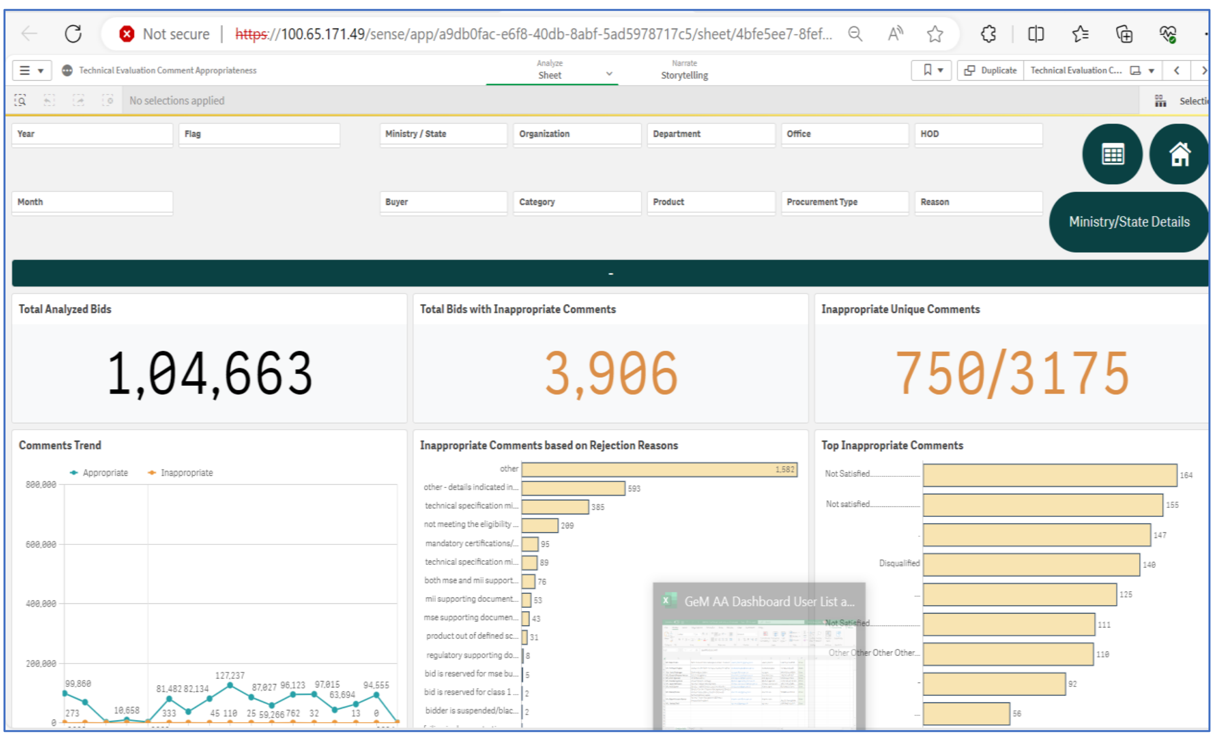
This solution is consumed by two ways, i.e. on portal at bidding process and through BI dashboard.

- Bidding process:
  When a rejection occurs, the buyer selects a reason from a drop-down list and adds comments. The GeM system should restrict inappropriate or useless comments during input to minimize irrelevant feedback from the start. This is achieved through API which processes comments and

applies ML model to identify relevant / irrelevant comment. On basis of outcome below is alert provided to buyer on quoting irrelevant comment.



- Dashboard: Dashboard gets updated monthly and comments with buyer and bid number details are showcased through dashboard. GeM validation team has the access to this dashboard.



Data:

To build this solution required data consist of Bid number, buyer Id, Category Id, product Id, Status, Reasons, Comments, Category, bid starting date, bid ending date and seller Id. The required data is fetched from below tables.

- Table: bdp_odb_bids
- Table: bdp_odb_bid_participation
- Table: bdp_odb_bid_participate_details

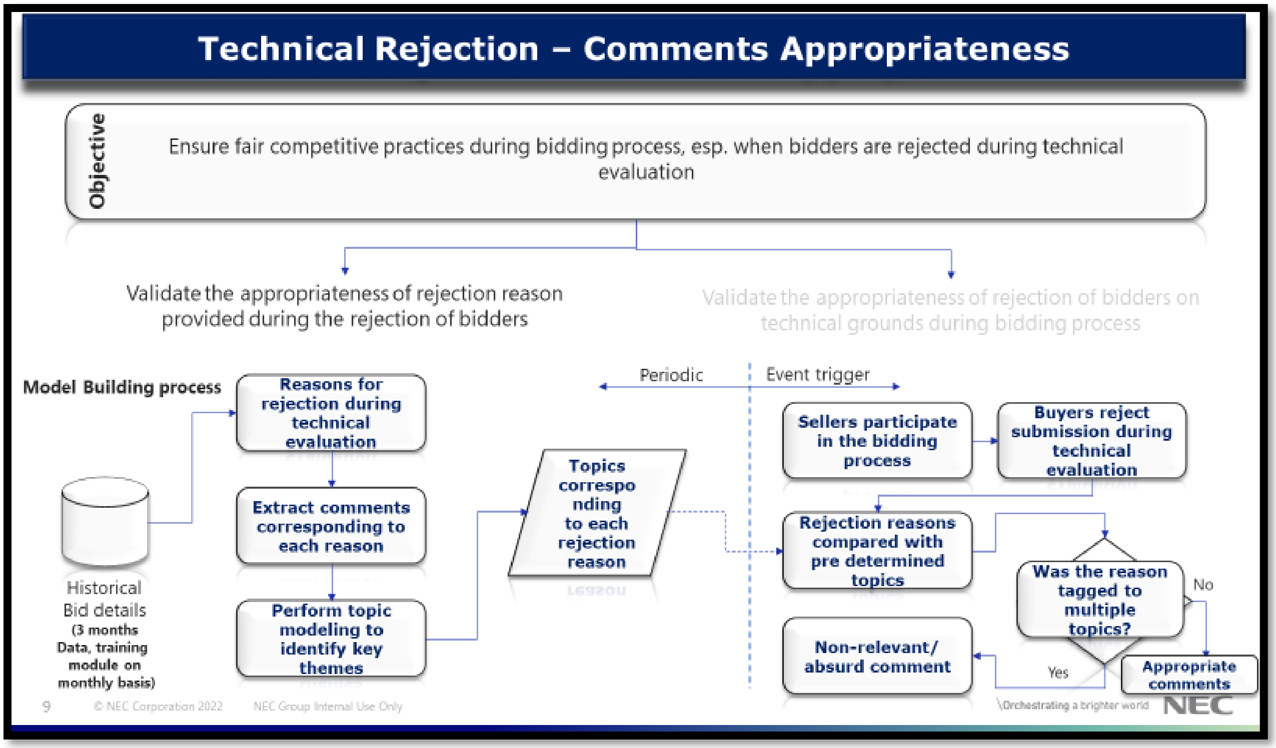
Model Methodology:

1. Data preprocessing:
    - It is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.
    - When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, data in json format converted to structured data.

2. Topic modeling:
    - It is a machine learning technique that automatically analyzes text data. These are very useful for the purpose for organizing large blocks of textual data. In natural language processing, Latent Dirichlet Allocation (LDA) is a generative statistical model that explains a set of observations through unobserved groups. It is popular topic modeling technique used to classify text into a particular topic.
    - The topics produced by topic modeling techniques are clusters of similar words. A topic model captures this intuition in a mathematical framework, which allows examining and discovering, based on the statistics of the words.
    - The probabilities are generated based on this topic values for each comment in reasons wise. These probability values are calculated, the differentiation is done between the maximum probability and minimum probability values.

3. Machine Learning algorithms
    - Many machine learning algorithms are capable of predicting a probability or scoring of class membership, and this must be interpreted before it can be mapped to a crisp class label.
    - This is achieved by using a threshold, where all values equal or greater than the threshold are mapped to one class and all other values are mapped to another class.

4. Thresholds:
    - There the threshold limiting of probabilities takes place at which the inappropriate comments will be having very low differentiation value.
    - So, by this process we can find out the comments that are not remarked correctly as per the rejection in the bidding.



Steps In Model:

1. Data Extraction:
   - Extract all necessary information from the database. Ensure the data contains comments in JSON format, as well as other relevant fields required for analysis.

2. Data Cleaning (Pre-processing):
   - Convert JSON data (containing reasons and comments) into a structured format like a Python DataFrame.
   - Handle missing values, duplicates, and standardize the text data for further analysis.

3. Importing Libraries and Initial Exploration:
   - Import required libraries (e.g., pandas, numpy, sklearn, spacy).
   - Load the dataset into a DataFrame and explore the different types of "reasons" present in the data.

4. Reason-wise Data Selection and Word Vector Embeddings:
   - Segment the data based on "reason".
   - Generate word vector embeddings for comments.
   - Create a Latent Dirichlet Allocation (LDA) model to uncover the hidden topics within each set of comments.

5. Saving Models:
   - Save the Count Vectorizer model and LDA model to disk using libraries pickle for later use.

6. Generating Probabilities for Comments:
    - For each comment, compute the topic distribution probabilities using the LDA model. This step helps in understanding the relevance of each comment to the identified topics.

7. Differentiating Probabilities:
    - Calculate the difference between the maximum and minimum probability values for each comment.
    - Map these values back to the DataFrame for further analysis.

8. Filtering with Thresholds:
    - Apply a threshold to filter out comments that have a low probability of belonging to any significant topic. These are considered irrelevant.

9. Predicting Irrelevant Comments for New Data:
    - Load the saved models (from the last 3 months) to process new incoming data.
    - Use these models to predict irrelevant comments based on the patterns learned from previous months' data.

Business Value:

1. The solution aims to enhance the bidding process on the GeM portal by identifying and filtering out irrelevant or inappropriate comments provided as reasons for bid rejections. By focusing on providing relevant and constructive feedback, the solution will improve the seller's experience by ensuring that rejection reasons are meaningful and actionable. This not only helps sellers better understand the shortcomings of their bids but also fosters a more transparent and fair bidding environment.

2. Additionally, the GeM management will benefit from this solution by being able to identify and address buyers who consistently provide inadequate or inappropriate feedback. By addressing these issues, GeM management can ensure that the bidding process remains rigorous and fair, ultimately leading to more accurate evaluations and increased product sales and business transactions. This refined approach to feedback and bidding will contribute to a more effective marketplace, benefiting both sellers and buyers alike.

XVIII. Technical Specification Rejections Validation

Business Requirement:

Currently, while bidding on the GeM portal, there is a challenge where some offered products are being rejected by buyers even though they meet the required specifications. There isn't a module to monitor such cases and ensure fair competition by preventing buyers from rejecting products that meet the necessary specifications.

This solution improves the process by providing a mechanism to compare all offered products against a base product in terms of technical specifications. The model will analyze and score all products, enabling buyers to make more informed decisions. This solution will help prevent the rejection of technically compliant products, thus enhancing the experience of sellers on the GeM portal and promoting fair and transparent bidding.

Proposed solution:

Develop Technical Specifications Comparison model with Product Similarity using AI/ML algorithms/ business rules. This Module will compare the products and its golden specifications when compared with base product.
The model will enhance buyer awareness, enabling more balanced and fair decision-making, and fostering a more transparent and efficient bidding environment on the GeM portal.
Note: This use case is not the part of the RFP as suggested by CEO sir NEC created the solution.

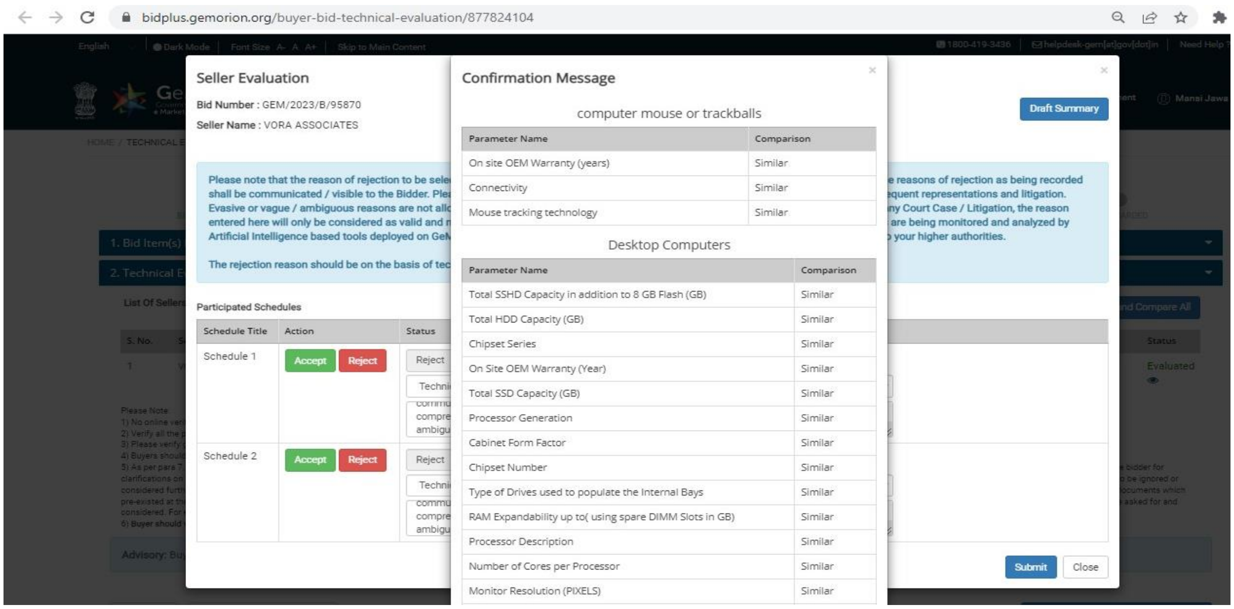
Solution Consumption:

Following are the stakeholders and processes where this solution can have impact.

- Buyers:
  - The solution provides buyers with enhanced awareness through an automated comparison system that evaluates all offered products against a base product's technical specifications. While buyers retain the ability to select any product, the solution will present them with a clear comparison of each product's specification compliance.
  - A "specification compliance score" will be generated for each product, which allows buyers to see how closely each offered product aligns with the base product's requirements. This transparency ensures that buyers are informed and can make more objective and data-driven decisions.

- Sellers:
  - Sellers will benefit from increased transparency in the evaluation process. Products that meet the required specifications will be clearly identified, reducing the likelihood of rejection due to misinterpretation or oversight by buyers.

The solution is consumed on GeM portal at bidding process. This module provides a detailed comparison of all offered products against the base product. Ensure that this comparison includes key specifications and features to aid the buyer’s decision-making.



During the technical Evaluation in bidding process when buyer will reject the product offering of any seller. A pop up will come on which buyer consent will be taken for further rejection of seller with an alert message and the product comparison details. Product similarity score shall also be shown on evaluation page.

Data:

To build this solution required data consist of Bid number, buyer Id, Category Id, product Id, Status, Reasons, Comments, Category, bid starting date, bid ending date and seller Id. The required data is fetched from below tables:

- Table: bdp_odb_bids
- Table: bdp_odb_bid_participation
- Table: bdp_odb_bid_participate_details

Model Methodology:

The model methodology focuses on the systematic transformation and analysis of data to enable precise machine learning applications. The process begins with the ingestion of raw product data and similarity scores, which are then processed to ensure compatibility with the model's requirements.

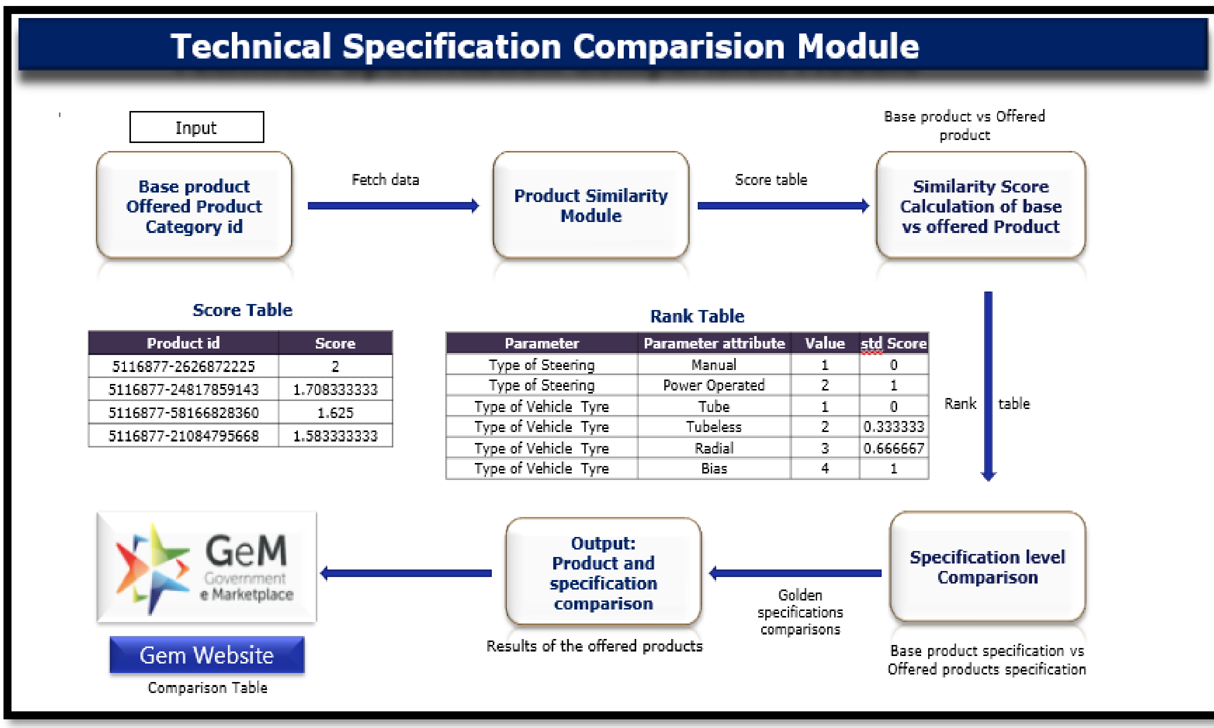
The core component of the methodology is the product similarity module, which provides score to products based on their feature vectors. Initially, the raw data undergoes a comprehensive processing and normalization phase, where numerical features are standardized, categorical variables are encoded, and text data is preprocessed. This normalization ensures that all data is in a consistent format, facilitating accurate analysis.

Subsequently, the model performs feature extraction, identifying and isolating relevant attributes from the product data. Using these features, similarity scores are computed between the base product and other offered products.

The computed similarity scores are then used to conduct a comparative analysis of each offered product relative to the base product. This analysis evaluates how well each offered product aligns with the base product's specifications, focusing on key attributes and golden parameters.

To provide comparison awareness, the model generates detailed insights into how each offered product compares with the base product. This is achieved through in-depth scoring and reporting mechanisms that highlight the differences and similarities between the products.

Finally, the model produces a comprehensive output that includes both product comparison scores and detailed specification comparisons. This output offers a thorough view of how each offered product measures up against the base product, delivering actionable insights that support informed decision-making.



Steps In Model:

1. Import Libraries:
   Load the necessary libraries for data handling, processing, and analysis.

2. Data Normalization:
   Transform the input data from JSON format into a structured, normalized form suitable for processing.

3. Retrieve Product Similarity Scores:
   Access the product similarity scores from the relevant module or data source. This includes fetching the score table and associated attributes.

4. Compute Product Comparison Scores:
   Calculate comparison scores between the offered products and the base product. Apply a similarity metric to determine how closely the offered products match the base product.

5. Retrieve Rank Table Specifications:
   Extract the rank table specification scores from the output of the product similarity module, which includes data on product rankings and specifications.

6. Extract Specification Data:
   Obtain raw specification data from the input of the product similarity module, ensuring all relevant product details are captured.

7. Perform Specification Comparison:
   Compare the specifications of the base product with those of the offered products using appropriate similarity metrics. Generate similarity scores based on the comparison.

8. Generate Final Output:
   Produce the final output that includes both product comparison scores and specification similarity comparison for all offered products in relation to the base product.

Business Value:

1. By providing detailed comparisons of offered products against the base product before any rejection, the solution ensures that buyers can make well-informed decisions based on accurate and comprehensive data. This approach enhances the fairness of evaluations by offering a clear understanding of how each product aligns with the required specifications. It reduces discrepancies and prevents unjust rejections, making the bidding process more efficient and effective.

2. Additionally, this methodology ensures that the GeM portal operates at its fullest potential, upholding the principles of free competition. By fostering transparency and fairness in transactions, it guarantees that all bidding processes are conducted equitably. This not only benefits both buyers and sellers but also promotes increased product sales and business transactions, supporting a robust and competitive marketplace.

XIX. Text Search

Business Requirement:

GeM requires a robust and scalable solution to efficiently search through large datasets containing product and seller information. The goal is to streamline the process of retrieving relevant product data based on specific keywords, improving operational efficiency. To meet this need, a keyword-based data search and retrieval system has been developed and deployed. This system allows stakeholders to quickly search, access, and export product information, thereby enhancing decision-making and data management.

Proposed solution:

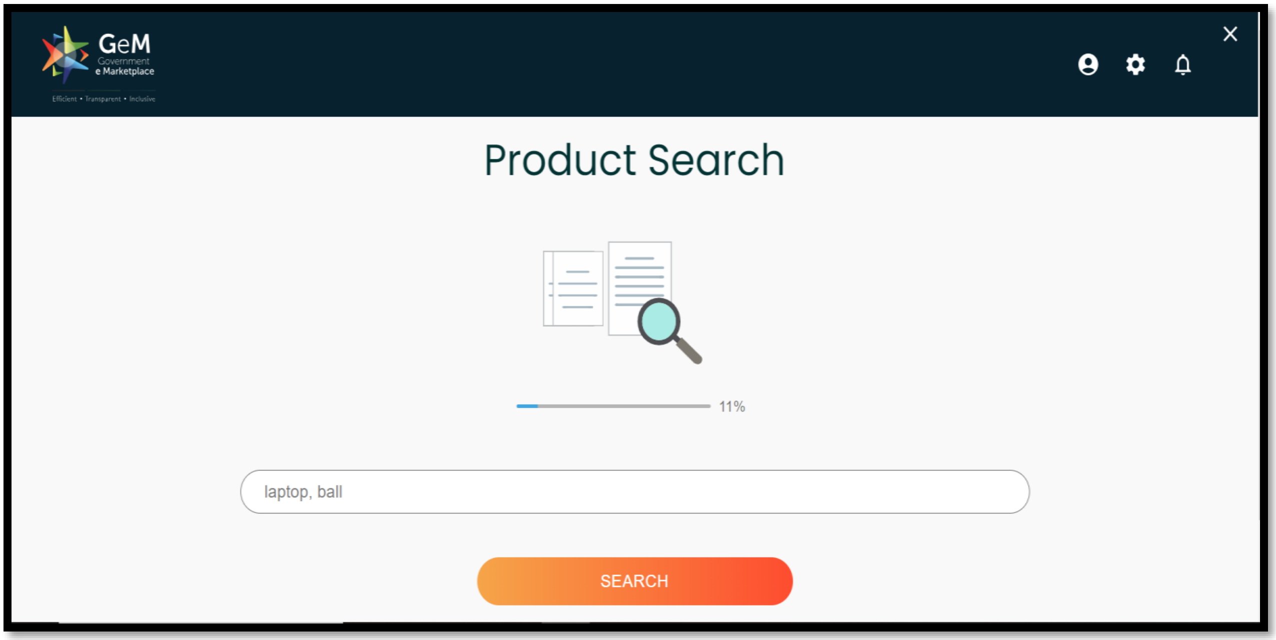
To address the challenge of managing and searching large datasets, a keyword-based data search and retrieval solution has been developed. This solution allows users to:

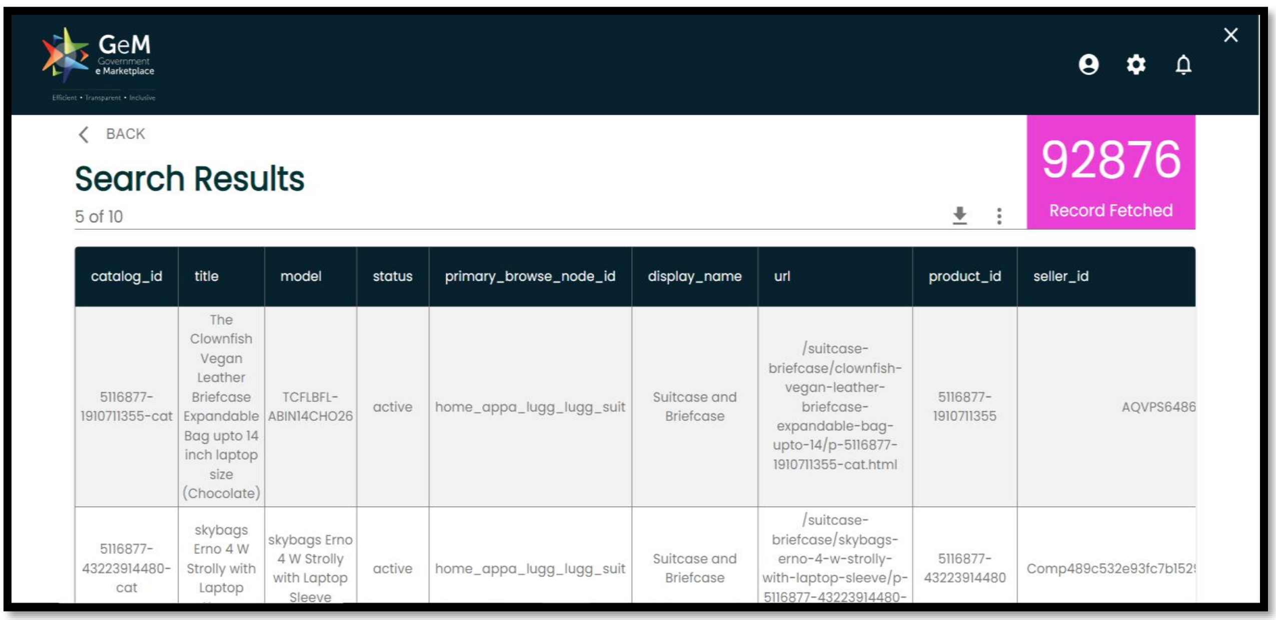
- Merge and process multiple CSV/Excel files concurrently to create a unified dataset.
- Perform keyword-based searches across relevant columns.
- View the top results from the search and download a full report of the results in CSV format.
- Provide a user-friendly web interface (UI) for easy interaction with the system.

This solution enhances the data retrieval process within GeM, allowing stakeholders to quickly search, access, and export the required information efficiently.

Solution Consumption:

This solution, designed to streamline the search and retrieval of product data, will be consumed by the Market Sanity Team and other GeM stakeholders**.**





Data:

The required data can be fetched from below tables to support keyword-based searches and data retrieval:

- Table: inb_variants
- Table: inb_browse_nodes
- Table: inb_stock
- Table: inb_sellers

Business Value:

1. Faster Data Retrieval: By enabling keyword-based searches across large datasets, this solution significantly reduces the time taken to retrieve product information, allowing the Market Sanity Team and other stakeholders to focus on core tasks rather than manual data searches.

2. Automation of Repetitive Tasks: Automated data merging and searching eliminate manual effort, which reduces errors, saves time, and increases productivity for stakeholders.

3. Real-Time Data Access: Stakeholders have immediate access to accurate and up-to-date product information, allowing them to make more informed decisions when validating products, procuring goods, or managing inventory.

4. Handling Large Datasets: This solution is designed to efficiently handle large volumes of data, making it scalable as the GeM platform continues to grow in terms of product listings and sellers.

XX. BOQ (Bill of Quantities)

Business Requirement:

There are two critical issues related to buyer bidding behaviour:

- Intentional Query Manipulation: Some buyers intentionally manipulate the query within the BOQ category to gain an unfair advantage or to misrepresent the product being procured.
- Unintentional Misclassification: Buyers who are not familiar with the exact product category might inadvertently place bids in incorrect categories.

To address this, GeM aims to develop a BOQ module that can intelligently analyze a buyer's query and suggest the top 5 categories that may correspond to the input query. This module will help in accurately categorizing buyer queries and ensure that they are directed to the most relevant categories available on the GeM portal.

Proposed Solution:

The BOQ module is designed to provide intelligent category recommendations for buyer queries in the bidding process. The solution will:

- Analyze the buyer's query/requirement.
- Provide the top 5 possible categories that best match the input query.

This module will significantly enhance the bidding process by ensuring that buyer queries are accurately mapped to the most relevant categories, leading to more precise and effective procurement decisions.

Solution Consumption:

Currently, the BOQ module solution generates detailed reports that are sent to GeM for review and validation. These reports contain the top 5 category suggestions for each query processed through the BOQ.

In the future, this solution will be directly integrated into the bidding process on the GeM portal. This will allow for real-time validation and category suggestions as buyers submit their queries, enhancing the efficiency and accuracy of the procurement process.

Data:

The necessary data for building this solution includes:

- Category Data: Title, Category Name, and Category ID.
- Bid Data: Item Title, Item Description, Bid Number.

The required data can be fetched from below tables:

- Table: bdp_bids
- Table: bdp_bid_details

Model Methodology:

The methodology for the BOQ module revolves around fuzzy matching and semantic similarity to identify the top categories relevant to a given bid query. Here's a breakdown:

1. Data Preparation:
   - Load the category information which contains details such as the category title, ID, and name.
   - Load the bid data which includes the item titles and descriptions.

2. Text Preprocessing:
   - Concatenate item titles and descriptions to create a unified representation (`merged`) for each bid query.
   - Use text preprocessing techniques, such as stopword removal, to clean and standardize the input data.

3. Fuzzy Matching:
   - Apply fuzzy matching using the `fuzzywuzzy` library to compare each bid query against the category titles.
   - The goal is to compute similarity scores between the query and each possible category.

4. Top 5 Category Selection:
   - For each bid query, the top five categories are selected based on their similarity scores.
   - These top matches represent the most relevant categories for the given bid.

5. Mapping and Refinement:
   - Map each bid query to the top five categories identified through fuzzy matching.

6. Brand Flagging:
   - Load a list of negative brand names.
   - Perform brand extraction and flagging by identifying and highlighting negative brand names within the bid descriptions.

7. Result Compilation:
   - Combine the results, including the bid numbers, top categories, and flagged brands, into a final DataFrame.
   - Save the results to an Excel file for further analysis or reporting.

Business Value:

1. Increased Efficiency in Query Processing: By automating the categorization of buyer queries, the BOQ module reduces manual effort and potential errors, leading to a more efficient bidding process.

2. Improved Transparency: The module aligns with GeM’s goal of enhancing transparency in public procurement by ensuring that buyer queries are matched with the most relevant categories, providing clear and accurate information for decision-making.

3. Detection of Negative Brands: The module’s capability to detect negative brands in buyer queries adds an additional layer of compliance and quality control, ensuring that only approved brands are considered in the bidding process