Building a Data Warehouse to store data on Ethiopian medical business data scraped from telegram channels

Week 7 Challenge Document

Date: 09 Oct - 15 Oct 2024

Table of Contents

1. Introduction:

This project revolves around gathering data from multiple sources, including web scrapers and Telegram channels, and integrating it into a centralized data warehouse to facilitate comprehensive business analysis. Additionally, object detection through YOLO (You Only Look Once) is incorporated to further enrich the data and unlock deeper insights.

Data warehousing plays a crucial role in enhancing the quality and accessibility of data for analysis. By consolidating fragmented data into a single location, businesses can perform more accurate, efficient, and extensive analyses. This centralized approach enables the identification of trends, patterns, and correlations that are difficult to discern with unstructured or distributed data sources. The ability to query and report from a well-architected data warehouse empowers businesses to generate actionable intelligence rapidly, providing a competitive edge in decision-making.

The success of this project depends on two key processes: ETL (Extract, Transform, Load) and ELT (Extract, Load, Transform). The ETL process extracts data from various sources, cleans and transforms it into a standardized format, and loads it into the data warehouse, ensuring data consistency and readiness for analysis. ELT, on the other hand, loads raw data into the data warehouse first and transforms it as needed, leveraging the computational capabilities of modern warehouses. These processes ensure seamless integration and transformation of the diverse data sets involved.

The project comprises five key deliverables:

- Developing a data scraping and collection pipeline to gather data on Ethiopian medical businesses from web and Telegram channels.
- Designing a data cleaning and transformation pipeline to ensure that the data is standardized and ready for analysis.
- Implementing YOLO for object detection, enhancing the dataset with valuable visual analysis capabilities.
- Building and implementing the data warehouse, enabling scalable and robust data storage.
- Integrating and enriching the data, combining the scraped data with additional sources for enriched insights.

By implementing these components, we aim to create a comprehensive data solution that supports deeper insights into the Ethiopian medical business landscape.

2. Data:

2.1 Data Scraping and Collection Pipeline:

1. Telegram Scraping: Utilize the Telegram API or write custom scripts to extract data from public Telegram channels relevant to Ethiopian medical businesses. Use the following channels
    - https://t.me/DoctorsET
    - Chemed Telegram Channel
    - https://t.me/lobelia4cosmetics
    - https://t.me/yetenaweg
    - https://t.me/EAHCI
    - And many more from https://et.tgstat.com/medicine
2. Image and Scraping: Collect images from the following telegram channels for object detection:
    - Chemed Telegram Channel
    - https://t.me/lobelia4cosmetics

Some of them were shown below:

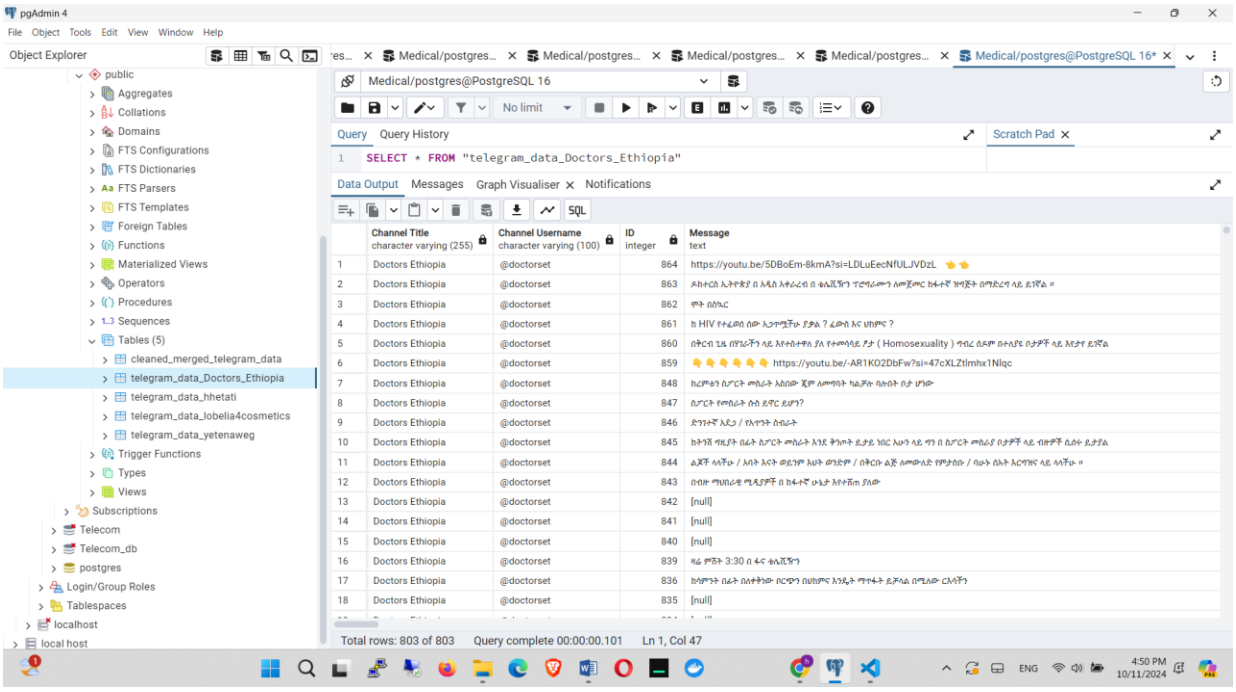


Figure 1: Telegram_channal_doctors_ethiopia (all other were putted in github link)

2.2 Data cleaning:

In the data cleaning process, ensuring the quality and consistency of the data is essential to maintaining the accuracy and reliability of downstream analysis. For the given function `clean_telegram_data`, the following cleaning stages and methods were implemented:

- Loading the Data:The cleaning process begins by reading the input CSV file into a pandas DataFrame. This ensures that all data is readily accessible for transformation and cleaning operations.
- Selecting Relevant Columns:The raw dataset may contain more columns than necessary for the project. In this step, only the relevant columns—"Channel Title", "Channel Username", "Message", "Date", and "Media Path"—are retained. This helps reduce data clutter and focuses on the columns needed for analysis.
- Duplicate Removal:Duplicated rows can result from data collection errors or repeated messages. To maintain the integrity of the dataset, duplicates are removed using `drop_duplicates()`. The function also tracks the number of duplicate rows removed, allowing the user to assess the impact of this cleaning step.
- Handling Missing Values:Missing values are often a common issue in raw data. In this case, the `dropna()` method is applied to remove any rows that contain missing values

across any column. This step ensures that the data remains complete, avoiding potential errors in later stages of analysis.

- Standardizing Text Formats: Consistent text formatting makes querying and analysis easier. The "Channel Title" column is standardized to title case using `str.title()`, ensuring uniformity across all values.
- Date Conversion and Validation: Dates are a critical component of time-series data, and inconsistencies in date formats can cause issues in analysis. The "Date" column is converted into a standardized datetime format using `pd.to_datetime()`.

Saving the Cleaned Data

Once the data has been cleaned, it is saved to a new CSV file at the specified output path. This final step ensures the cleaned data is available for subsequent processing or analysis. The output shows in Figure 2.

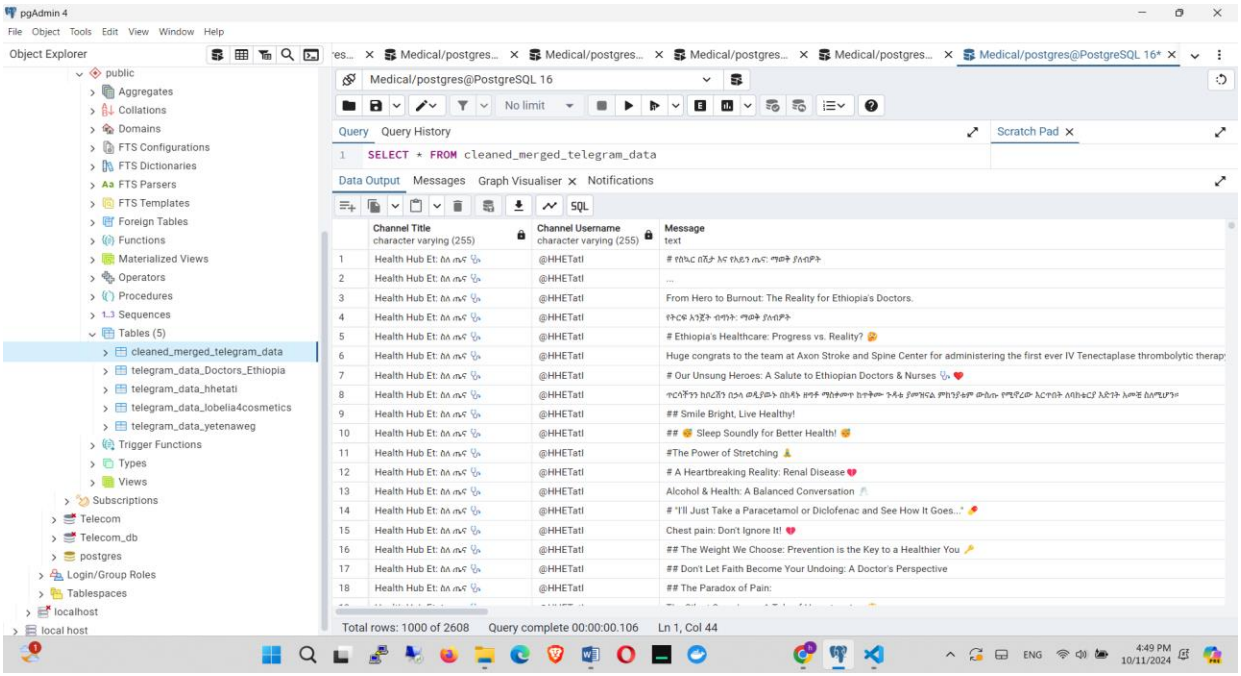


Figure 2: cleaned and merged telegram_channals

2.3. Transformation Pipeline:

The input data is sourced from the table `telegram_data_lobelia4cosmetics`

which contains raw message logs, with a focus on extracting two specific fields of interest:

- Product Name: Typically an alphanumeric string representing the name of a product, extracted from the beginning of a message.
- Usage Information: Numeric data, potentially accompanied by additional text (e.g., units or descriptions), representing usage metrics or values found later in the message.

2.3.1 Transformation Logic

The transformation process can be broken down into two main stages: extraction and filtering.

2.3.1. 1 Extracting Product Name

Objective: To extract a product name from the beginning of the `message` field. A regular expression is used to identify strings that begin with alphabetic characters or spaces (`^[A-Za-z]+`). This pattern matches cases where the message starts with a product name. Once a match is identified, the product name is extracted using the `SUBSTRING()` function with the pattern `^(.*?)\s+`, which captures the first sequence of letters before the first space. The extracted product name is then trimmed off any leading or trailing spaces using the `TRIM()` function.

2.3.1.2 Extracting Usage Information

Objective: To extract numeric usage data and any associated text from the `message` field.

Another regular expression pattern (`\s[0-9]+\s?[A-Za-z]*`) is used to search for numeric values followed by optional text. This pattern is designed to capture numbers (e.g., usage values) and optional descriptions or units.

The final output of the transformation consists of the following columns:

- id: The unique identifier for each message.
- product_name: The extracted product name from the message, or `NULL` if no match is found.
- usage_info: The extracted usage information from the message, or `NULL` if no match is found.
- date: The date associated with the message.

The transformation is materialized as a table, providing a persistent dataset that can be queried for further analysis or reporting purposes.

This transformation method effectively processes unstructured text messages to extract key information, such as product names and usage data, in a structured format. By employing regular expressions and conditional logic, the process filters and outputs only meaningful rows, enabling more refined and actionable insights. If it is visible Figure 3 and 4 shows the results.

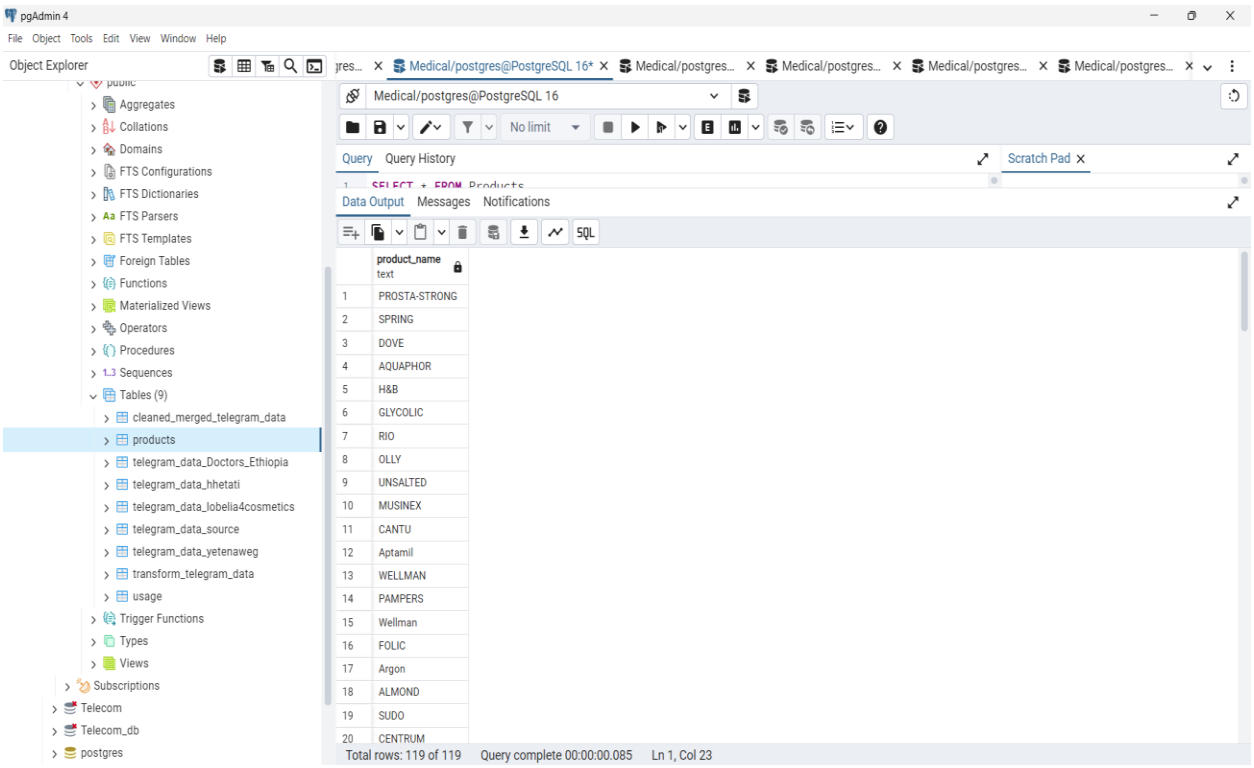


Figure 3: Product name

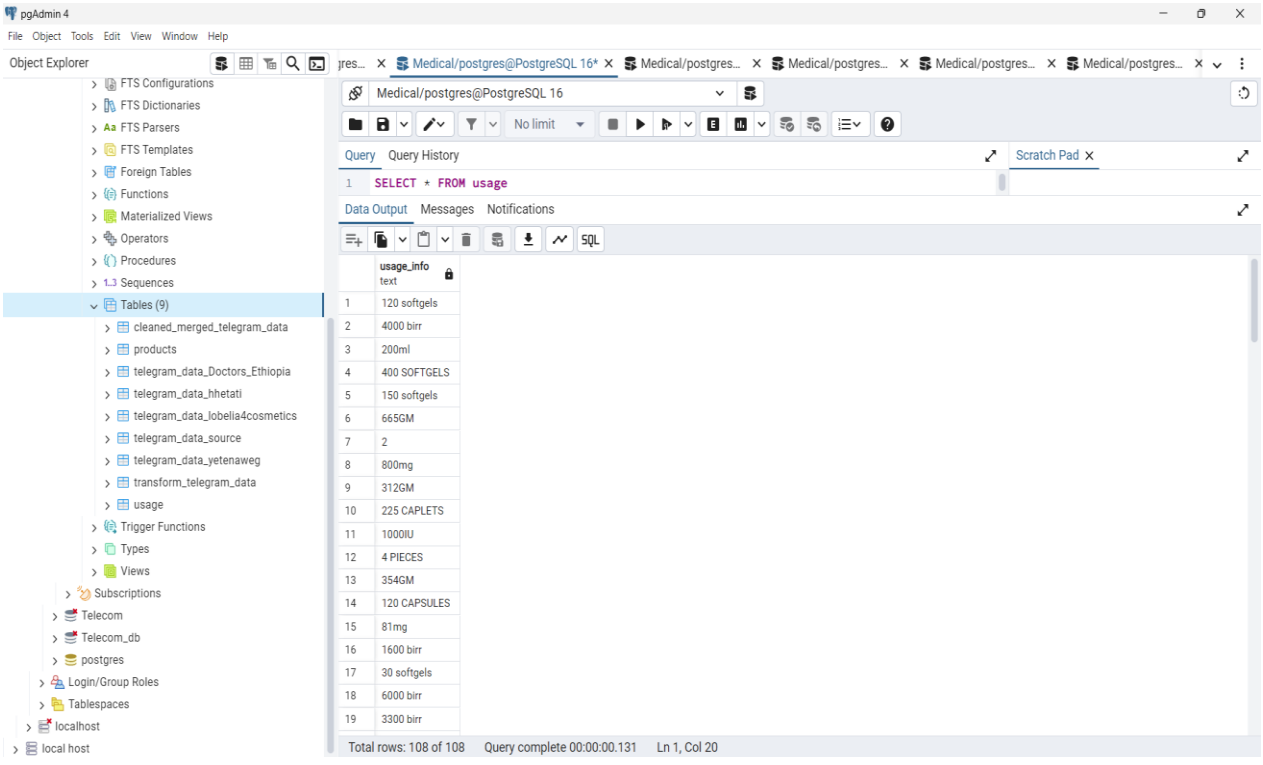


Figure 4: Usage labled

3. Object Detection Using YOLO

3.1 Using pre-trained Yollo5 Model

For this I used two datas:

1st: DoctorsET

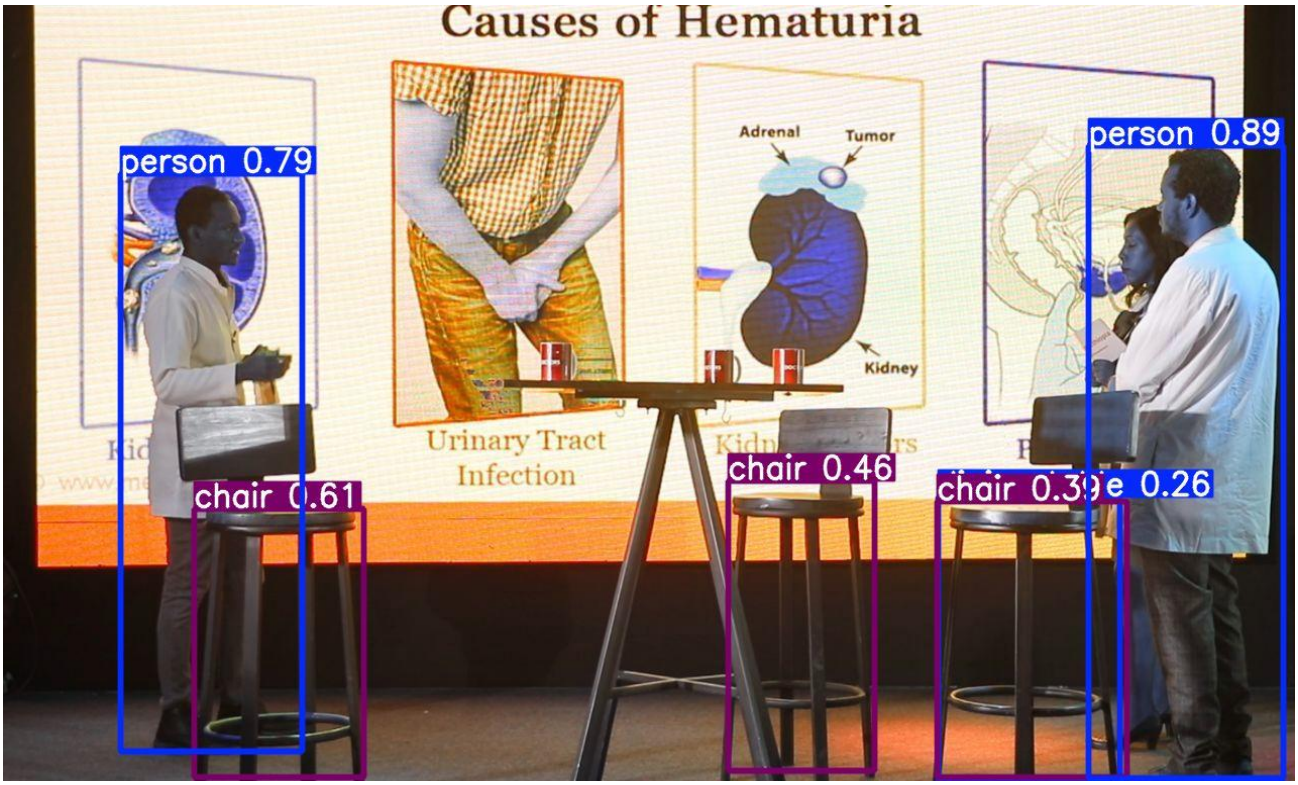
2ND: Lobelia4cosmetics

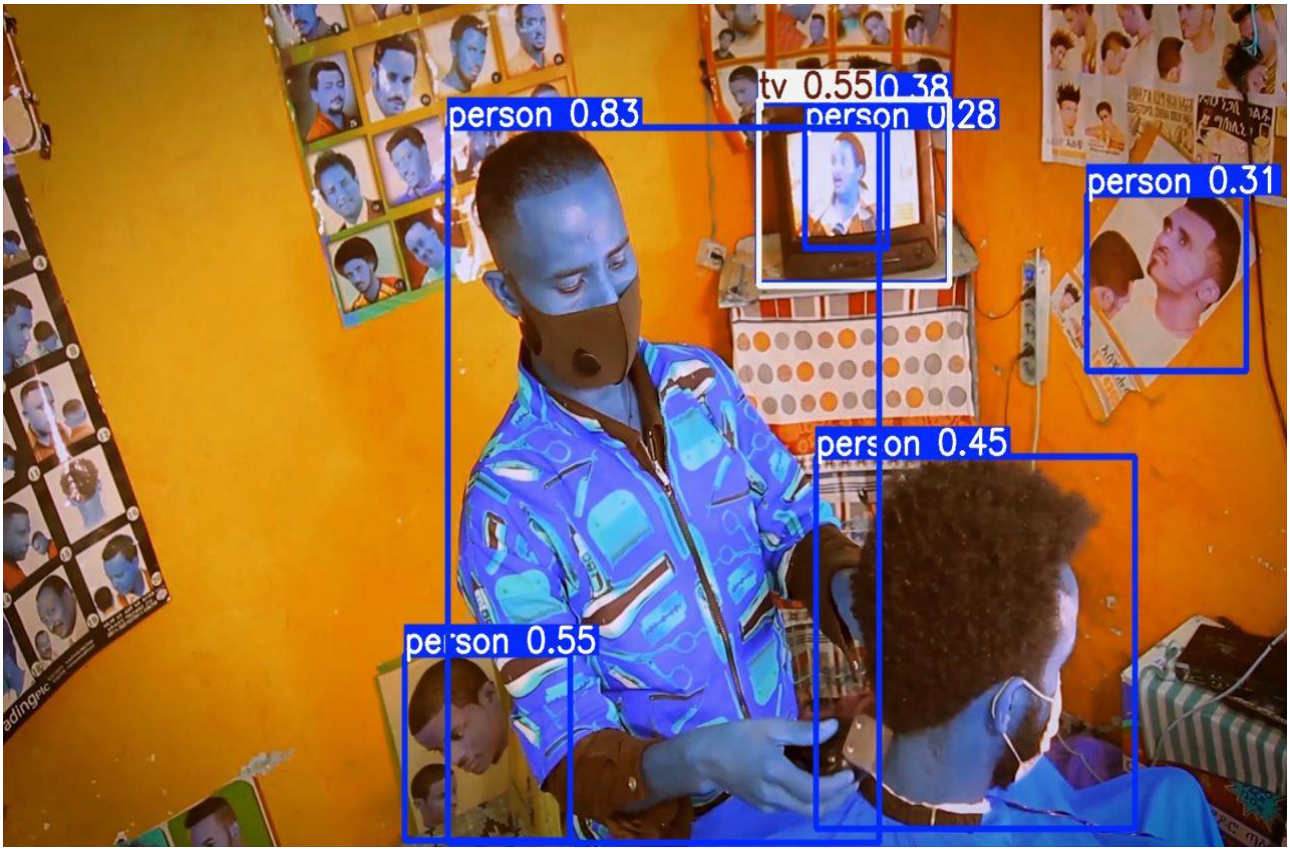
This is done by following this steps:

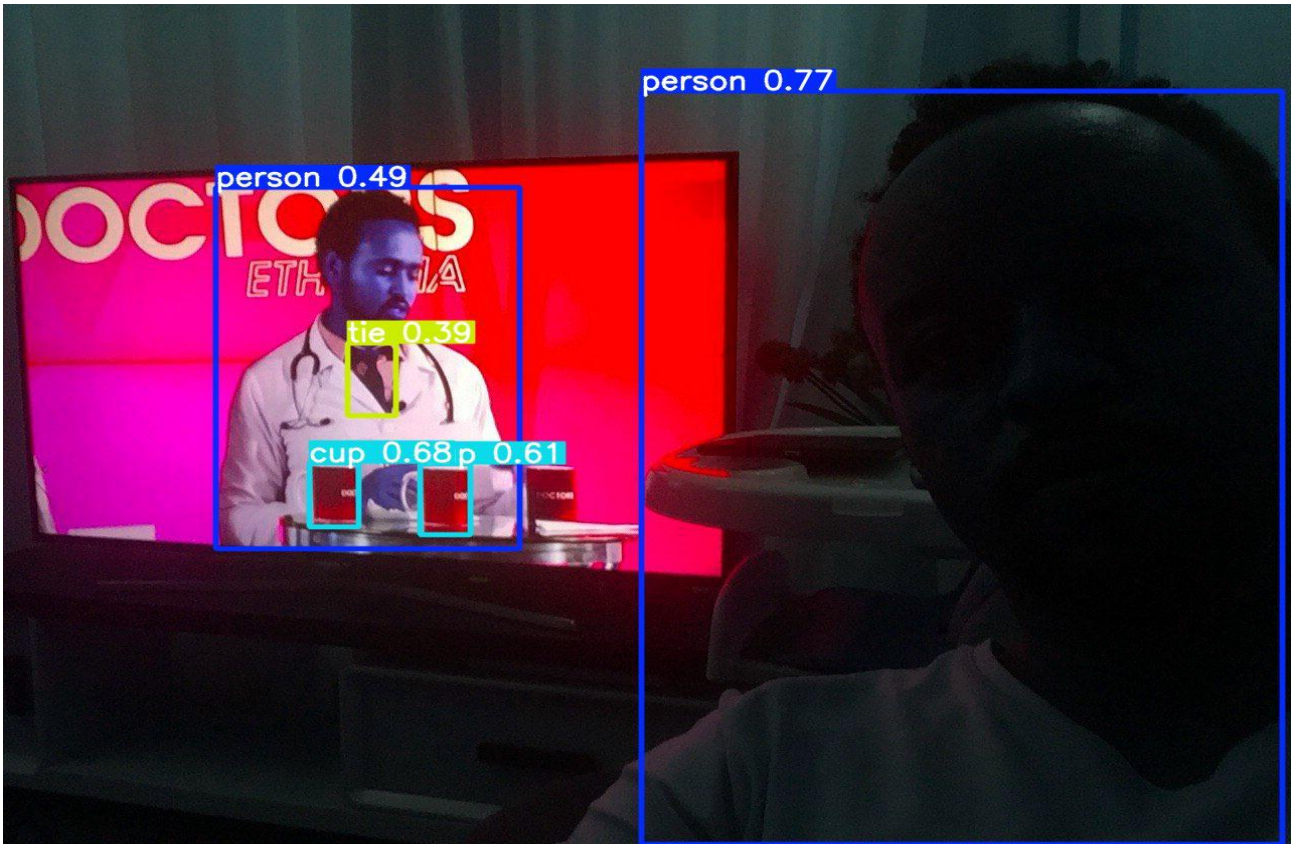
Model Loading: It loads the pre-trained YOLOv5 model ('yolov5s' version) using PyTorch's `torch.hub` feature, which enables easy access to the YOLOv5 model repository.Image Directory: The script specifies a folder path (`image_folder`) where the input images are stored. It targets files with `.jpg` and `.png` extensions.Image Processing:The script iterates through all the files in the folder.It verifies whether the file has a valid image format (either `.jpg` or `.png`). For each valid image, it loads the image using OpenCV' s `cv2.imread()` function.If an image fails to load, an error message is printed, and the script skips that file.

Object Detection: The pre-trained YOLOv5 model is used to perform object detection on each image. This generates results that include the detected objects, their confidence levels, and bounding boxes. Visualization and Saving: Detected objects are visually displayed on the images using the `results.show()` function.The detection results (with annotated images) are saved to the specified output directory (`detection_results/`).

As per shown in the below figures This model demonstrates strong performance in detecting objects such as humans, cups, ties, chairs, and bottles, showcasing its capability in identifying common items typically found in everyday settings. *However, the primary objective of our project is to focus specifically on identifying medical equipment and related apparatus. While the current model excels at general object detection, it may not fully meet our needs for recognizing more specialized medical tools and devices, which are essential for our task.* Therefore, further fine-tuning or adapting the model to better suit medical-related detection may be necessary


Causes of Hematuria
Adrenal
Tumor
Kidney
Urinary Tract Infection
person 0.79
chair 0.61
chair 0.46
person 0.89

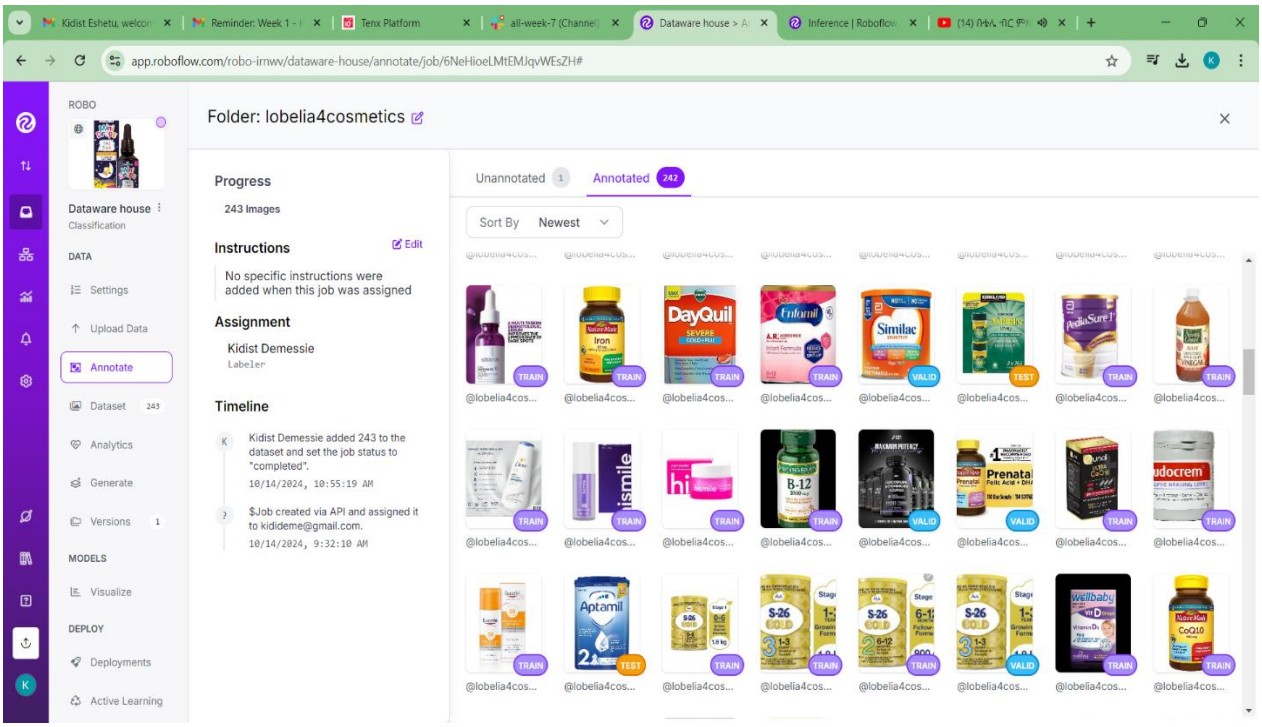

person 0.83
person 0.31
person 0.45
person 0.55

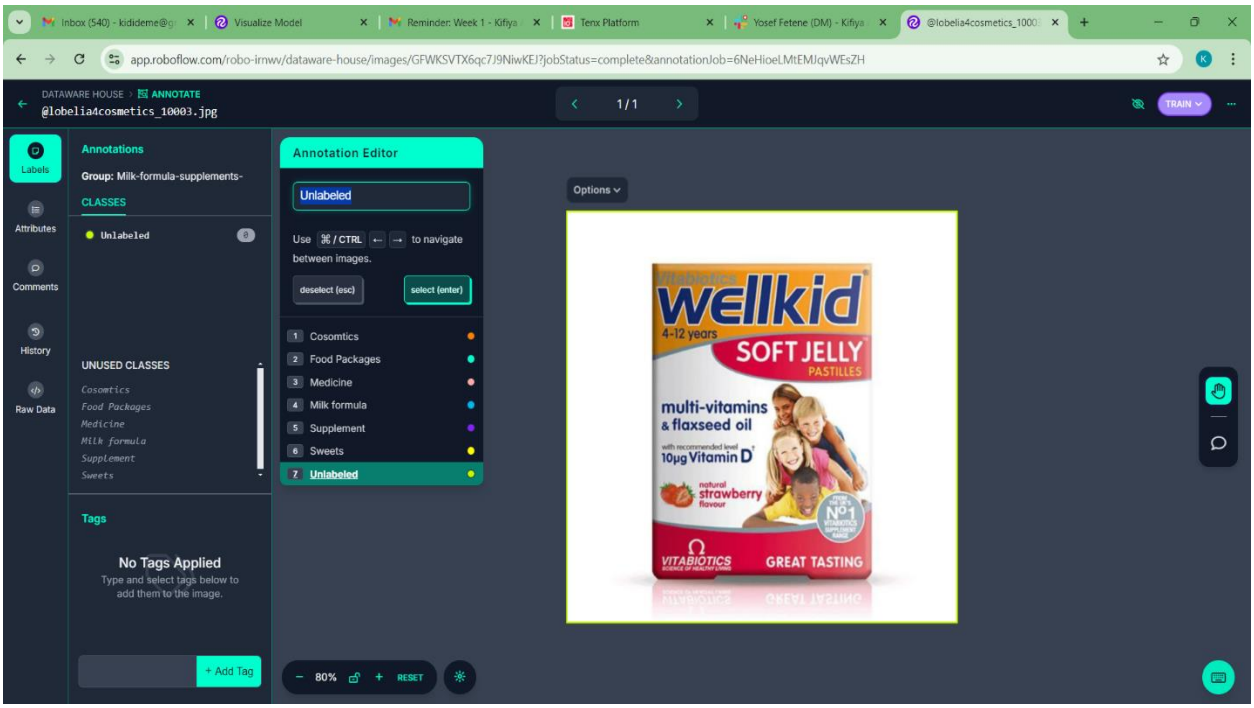

person 0.77
person 0.49
tie 0.39
cup 0.68
p 0.61


H&B
BOOKS & GOODS
MAGNESIUM
375MG
MUSCLE CARE
Supports healthy bones & muscles
and reduces tiredness & fatigue
VEGAN
180
1 A DAY
TABLETS

3.1 Using pre-trained Yollo8m Model

Before handling the data in YOllo8 we preprocess the data in roboflow (https://app.roboflow.com).This software helps in labeling the data in 7 different classes. The next dashboard shows the process:





Next,

1. Training Configuration

- Model: YOLOv8 medium (`yolov8m.pt`).
- Dataset:Located at `/content/datawarehouse_medical-1/data.yaml`, likely configured for detecting objects in medical images.
- Epochs: The model will be trained for 100 epochs.
- Batch Size: 16 images per batch.
- Image Size: 640x640 pixels.
- Automatic Mixed Precision (AMP): Enabled to improve training speed by using lower precision calculations while maintaining accuracy.

2. Model Architecture

- Layers: The model consists of 319 layers with a total of 23.2 million parameters.
- Computational Complexity: The model requires 67.9 GFLOPs (giga floating-point operations per second).
- Classes: The number of output classes has been adjusted from the default 80 (COCO dataset) to 10, to match the current dataset' s requirements.

3. Pre-trained Weights: The model has successfully transferred 433 out of 511 parameters from pre-trained weights. This helps accelerate the training process by leveraging learned features from a previous task.

4. Data Augmentation: The Albumentations library is used to apply various augmentations to the training images, including: Blur, Median Blur, Grayscale conversion,CLAHE (Contrast Limited Adaptive Histogram Equalization).These augmentations are designed to improve the model's generalization capability.

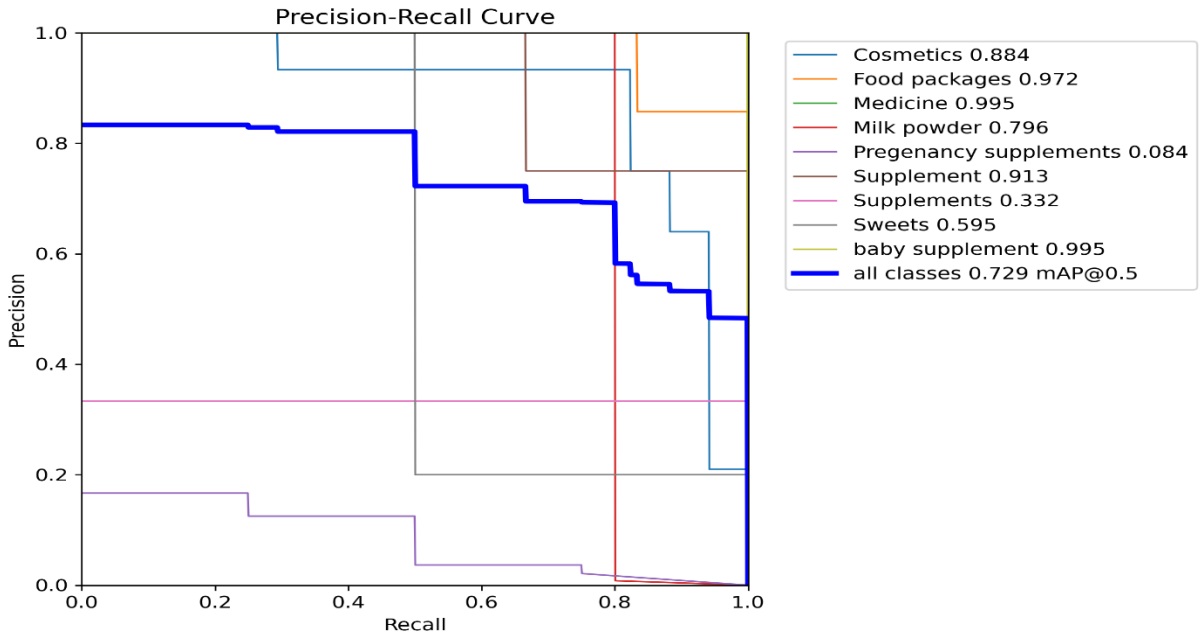
5. Optimizer: The optimizer was automatically determined to be AdamW with the following parameters:

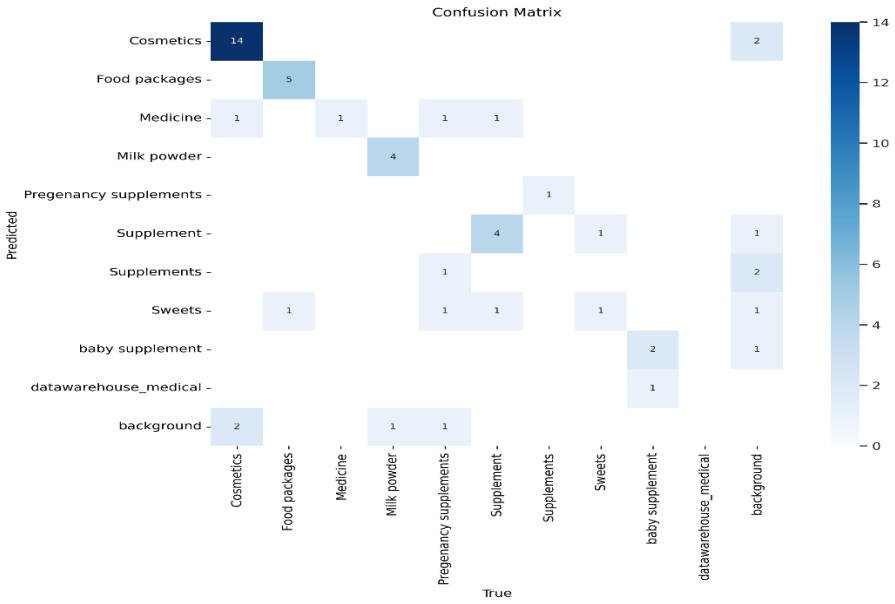
- Learning rate: 0.000714
- Momentum: 0.9
- Weight decay: 0.0005 for weights and 0.0 for biases.
- This optimizer configuration overrides the initial manual settings of `lr0=0.01` and `momentum=0.937`.

6. Validation Data: The model will be validated on a dataset of 49 images (including 4 background images), and no corrupt files were found.7.

The YOLOv8 model is now configured and ready for training on the medical dataset. The training includes using transfer learning with pretrained weights, data augmentation, mixed precision optimization, and AdamW as the optimizer. The following figure shows result of Precision-Recall curve with respected result.

- Best Performance: The "Medicine" and "Baby Supplement" categories performed exceptionally well with mAP50 values of 0.995 and 0.995, respectively, and strong recall.
- High Precision & Moderate Recall: The "Food Packages" class had a high precision of 0.979 but slightly lower recall at 0.833, suggesting occasional misses in detecting relevant objects.
- Low Performance: The "Pregnancy Supplements" and "Supplements" categories showed poor results, with mAP50 values of 0.084 and 0.332, respectively. Precision and recall in these categories are low, highlighting significant challenges in detecting these objects.
- Balanced Performance: "Cosmetics," "Milk Powder," and "Sweets" showed balanced performance but still had room for improvement, especially in recall for "Cosmetics" and "Milk Powder."





The model trained and predicted as a batch of 16 images . The result of tests looks like the following:

Figure: Batch 1

Figure: Batch 2

4. Key Findings:

Data Integration and Warehousing: Centralizing data from multiple sources, including web scrapers and Telegram channels, into a data warehouse improved data accessibility and analysis efficiency. This consolidation enabled businesses to identify trends and insights that were previously challenging to uncover with fragmented data sources.

Data Cleaning Process: The data cleaning pipeline, focusing on Telegram data, ensured data consistency and accuracy through stages like duplicate removal, handling missing values, and standardizing formats. This significantly improved the quality of the data used for analysis.

Data Transformation: These transformations ensured a more structured, cleaner, and consistent dataset for both product and price variables, enabling better analysis and model training.

YOLOv5 & v8 Results:

- Object Detection Performance: YOLOv5 was effective in detecting some key objects within the dataset, but its performance was not as strong as YOLOv8. It performed adequately for general object detection but struggled with finer categorization, especially for medical-related products like "Medicine" and "Baby Supplements."

- Class Detection Accuracy: YOLOv5 faced difficulties in detecting certain product classes like "Pregnancy Supplements" and "Supplements," leading to lower detection accuracy in these categories. Its ability to differentiate between similar product categories was less refined compared to YOLOv8.
- Model Efficiency: Despite being slightly faster than YOLOv8 in processing time, YOLOv5' s overall accuracy in terms of precision and recall was lower. This made YOLOv8 the preferred model for detailed analysis where higher accuracy was required, especially for detecting nuanced medical products.

Reference:

https://t.me/DoctorsET

Chemed Telegram Channel

https://t.me/lobelia4cosmetics

https://t.me/yetenaweg

https://t.me/EAHCI

https://app.roboflow.com