Health Data Science Hackathon Documentation

Background

In sub-Saharan Africa, HIV/AIDS is a leading cause of death, with women and children being the most affected. To combat the spread of HIV, governments and non-governmental organizations have implemented various programs, including Prevention of Mother-to-Child Transmission (PMTCT) programs. These programs aim to prevent the transmission of HIV from mother to child during pregnancy, childbirth, and breastfeeding.

However, there is a significant challenge in the reporting of PMTCT tests, with some facilities not reporting these tests. This inconsistency in reporting affects the accuracy of estimates of HIV prevalence, which is critical for planning interventions to reduce HIV transmission.

As a team that is passionate about solving Africa's most pressing problems we are working to solving this problem. Using our expertise in data science and machine learning, we are **developing a classification model to assist in the identification of PMTCT sites that do not report tests**.

Our solution involves analyzing datasets to identify different data behaviors, advise on the estimates where data gaps are found, and automate the clean-up of the data. With our approach, we are confident that we will provide accurate estimates of HIV prevalence, which will be crucial in the fight against the spread of HIV in the greater Sub-Saharan Africa.

To achieve this goal, we will analyze HIV testing data from select facilities in Kenya. We will use machine learning and statistical modeling to identify patterns in the data and to predict which facilities are most likely to have data gaps. Our approach will be validated by comparing the predicted data gaps with the actual data gaps identified by domain experts. By doing so, we will provide a data-driven approach to identifying facilities that are not reporting PMTCT tests, which will be a valuable tool for governments and NGOs to target their interventions.

We have faith that our strategy will significantly advance the fight against HIV in sub-Saharan Africa. We will assist governments and NGOs in planning and implementing more successful interventions by increasing the accuracy of HIV prevalence estimates, thereby lowering the rate of HIV transmission from mother to child.

A note of caution : This analysis is entirely backward-looking, relying on past performance data. This though is valuable information to help answer the questions and solve our problem statement above. At the same time, it shouldn't simply be extrapolated into the future. The observation period of the data (2019-2020) is relatively short, arguably not capturing a full program implimentation cycle. There probably has been an increase or decrease in the uptake of ART among pregnant women in ANC. Similarly, ART initiation among HIV-positive pregnant women might have also either increased from the records in 2020. Viral load suppression among PMTCT clients might have also evolved in different ways. With the above, we now understand that the behavior might differ significantly in the future. Thus before taking the final investment decision we recommend complimenting these findings with further forward-looking analysis.

Objectives

The main objective of this report is to analyze the presented datasets based on the reporting of the PMTCT tests using summary statistics, visualizations, statistical models, and textual explanations. It specifically seeks to:

1. Identify the different data behaviors in our datasets.
2. Understand the relationship between the PMTCT tests reporting and HTS test Reporting, then proceed to advise on the estimates where data gaps are found.
3. Provide a basis and guide the automation of the data cleaning process.
4. Dive deep into the step by step approach taken in the development of a classification model to assist in the identification of PMTCT sites that do not report tests.

Data Used

The dataset used in the analysis consists of a Four xlsx files, where each row represents a test report from a specific facility.

Two of the xlsx files contain three sheets from Migori ,Kakamega and Bungoma counties, categorized in their specific counties.
The following list below describes each sheet and its variables:

1. PMTC-KSM
   This sheet contains records of all reported HIV/AIDS tests done on pregnant women across different periods of the year 2019 and 2020 in Migori County, Kakamega County and Migori County.

   The following are all the attributes describing the sheet:
   - “Facility”: name of the facility
   - "facilityuid”: facility unique identifier.
   - “code”: a numerical identifier for each facility.
   - "ward”: the division of a sub-county.
   - “sub-county”: county divisions.
   - "county”: sub-regional divisions.
   - “indicator”: an identifier unique for a batch of tests.
   - “Indicator Name”: a human-readable indicator for tests
   - “period”: length of time in which reporting was done.
   - “dhis2_value”: number of tests reported to the DHIS system per facility.
   - “datim_value”: the aggregate for the number of tests reported to NGOs and partner organizations.

2. HTS_TST
   This file contains records of all reported HIV/AIDS tests across different periods of time by facilities in Migori County, Kakamega County and Bungoma.

   The following are all the attributes describing the sheet:
   - “Facility”: name of the facility
   - "facility”: facility unique identifier.
   - “code”: a simple and unique numerical identifier for each facility.
   - "ward”: the division of a sub-county.
   - “sub-county”: county divisions.
   - "county”: sub-regional divisions.
   - “indicator”: an identifier unique for a batch of tests.

- “Indicator Name”: a human-readable indicator for tests
- “period”: length of time in which reporting was done.
- “dhis2_value”: number of tests reported to the DHIS system per facility.
- “datim_value”: an aggregate of the number of tests reported to NGOs and partner organizations per facility.

The other two xlsx files differ in that they have test indicators of different counties in one sheet comapred to the other two that had only data from 3 counties and seperated into 3 different sheets.

The following list below describes each sheet and its variables:

1. HTS_comparison21

    - “facility”: name of the facility
    - "ward”: the division of a sub-county.
    - “sub-county”: county divisions.
    - "county”: sub-regional divisions.
    - "MOH_FacilityID”: facility unique identifier.
    - “MOH_IndicatorCode”: a numerical identifier for each facility.
    - “indicators”: a human-readable indicator for tests
    - “khis_data”: number of tests reported to the KHIS system per facility.
    - “datim_value”: the aggregate for the number of tests reported to NGOs and partner organizations.
    - “difference” : the difference in values between the khis data vaue and dataim data value.

2. PMTCT_comparison21

    - “facility”: name of the facility
    - "ward”: the division of a sub-county.
    - “sub-county”: county divisions.
    - "county”: sub-regional divisions.
    - "MOH_FacilityID”: facility unique identifier.
    - “MOH_IndicatorCode”: a numerical identifier for each facility.
    - “indicators”: a human-readable indicator for tests
    - “khis_data”: number of tests reported to the KHIS system per facility.

- “datim_value”: the aggregate for the number of tests reported to NGOs and partner organizations.
- “difference” : the difference in values between the khis data vaue and dataim data value.

Data Acquisition

Source Systems:

The data for this project was obtained from various health facilities in the country. The data was collected through the Health Information Management System (HIMS) and the District Health Information System (DHIS). These systems are widely used in the healthcare industry to collect, store, and manage patient data.

Data Acquisition Process

Data Gathering: Gathering data from the source systems was the first phase in the data acquisition process. The information was accessed by an allowed user in the Health System who is allowed access to the systems and shared to us in form of an excel format.

Data Extraction: After the data had been shared with us we gathered it by downloading it to our machines. To make using the data during the data analysis process easier, the data was exported to pandas dataframe .

Data Cleaning: To make sure the data was of the highest quality and prepared for analysis, the data underwent cleaning before being loaded into the working environment using pandas. Checking the data for duplicates, missing numbers, and inconsistencies was part of the cleaning procedure.

Data Loading: Following data cleansing, pandas was used to load the data into the working environment. Data frames, a type of pandas data structure that facilitates effective data manipulation and analysis, were used to load the data.

Data Preparation: The data was further prepared for analysis after it had been imported into the data frames. This required changing column names, eliminating extra columns, and converting the data's data types.

Data Analysis: When the data had been loaded and formatted, data analysis was the following stage in order to find trends, patterns, and insights. Data visualization, machine learning, and statistical methodologies were all used to accomplish this.

Exploratory Data Analysis

The data entails Kakamega and Migori counties HTS data and PMTCT data separately that were merged. The next stage was data cleaning.

Data Cleaning

1. Identifying and removing null values

This was done by using the info() functions to see if there were any missing entries per field. In the HTS data, only the datim_value" column had null entries of 111 rows while in the PMTCT data, all the columns had missing entries of 206 to 219 rows. In HTS data we filled in the null entries with zero while in PMTCT data we dropped the null entries.

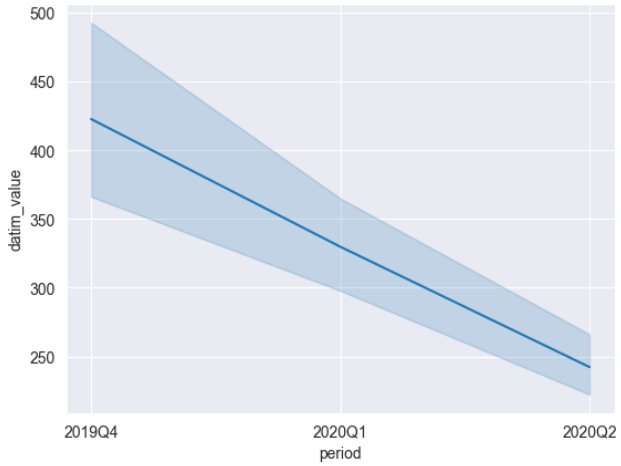
2. Extracting data in the "Indicator_name" column

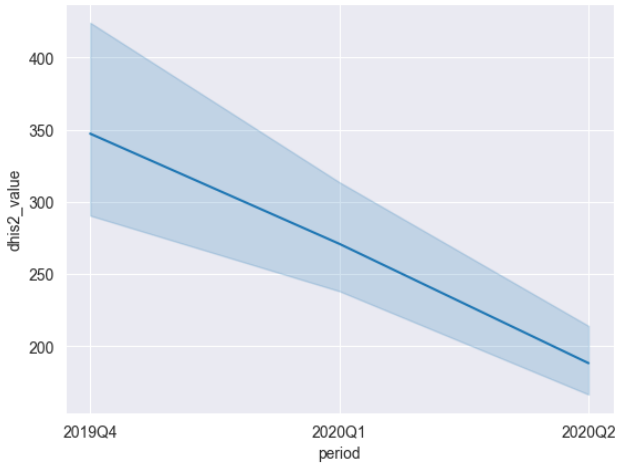
Extract the age and gender from the column into different columns and dropping the "indicator_name" column because all the information was the same.

3. Data manipulation

- Visualize datasets to try and identify the relationships between the columns.

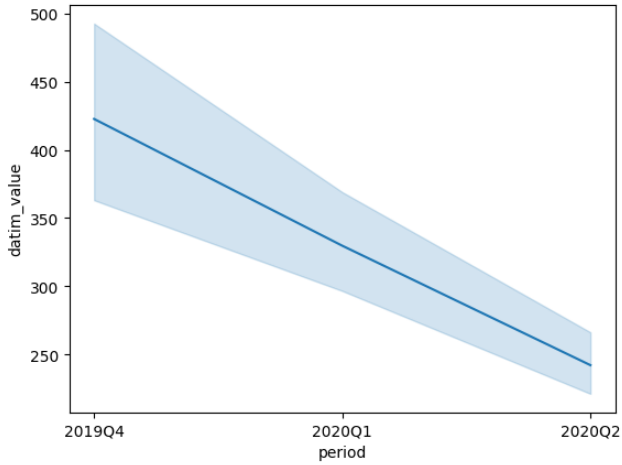
HTS Data

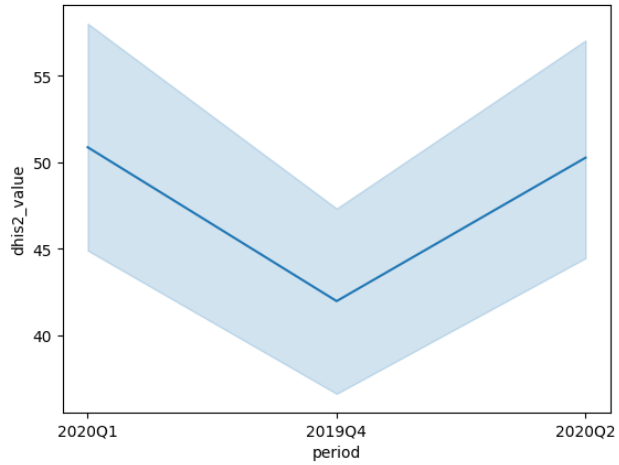




We can observe that the number of DHIS and DATIM value reports decrease with each subsequent quarter in HTS data

PMTCT Data





We can observe that the number of DHIS values decreases from the first quarter till 2019Q4 then increases while the DATIM value reports decrease with each subsequent quarter in PMTCT data

- Calculate the error difference between DHIS data and Datim data

There are facilities that had a difference in the DHIS and Datim in the HTS data. We first grouped them according to facilityuid and the period then calculated the total DHIS and Datim data respectively. After this, the first percentage difference is calculated and elimination was done using the median of the p

Feature Engineering

To perform feature engineering for the project above, we started by analyzing the available attributes in both datasets and selecting those that are most relevant for predicting HIV prevalence. Below is a description of the potential features that we derived from the given data:

1. PMTC-KSM:

- Number of reported tests by facility: This feature can be derived by summing up the `dhis2_value` and `datim_value` attributes for each facility. This would give us the total number of tests reported by each facility, which can be an important factor in predicting HIV prevalence.
- Test positivity rate by facility: This feature can be derived by dividing the number of positive tests reported by each facility by the total number of tests reported. This would give us the percentage of tests that were positive, which is a strong indicator of HIV prevalence.
- Facility location: We can extract the `ward`, `sub-county`, and `county` attributes to obtain the location of each facility. This can be useful in identifying regions with high or low HIV prevalence, which can inform targeted interventions.

2. HTS_TST:

- Number of reported tests by facility: Similar to the PMTC-KSM dataset, we can sum up the `dhis2_value` and `datim_value` attributes for each facility to obtain the total number of tests reported.
- Test positivity rate by facility: This feature can also be derived by dividing the number of positive tests reported by each facility by the total number of tests reported.
- Facility location: We can extract the `ward`, `sub-county`, and `county` attributes to obtain the location of each facility.

Additionally, we can considered the following potential features that can be derived by combining the data from both datasets:

- Total number of reported tests by location: We can aggregate the number of reported tests by location (i.e. ward, sub-county, and county) to obtain the total number of tests reported in each region. This can be useful in identifying regions with high or low testing coverage, which can inform targeted interventions.
- Test positivity rate by location: Similar to the previous feature, we can aggregate the number of positive tests by location and divide it by the total number of tests to obtain the test positivity rate for each region.

Model Development

1. Algorithm selection

For the model creation, the output is a binary classification model predicting either "yes" or "no" using the target variable "accept". The selected machine learning algorithm was the Decision Tree classifier because we are trying to classify between two outputs. We used the sklearn library.

2. Convert categorical data to numerical data and feature selection

We converted the categorical data in the 'period' and 'accept' columns to numerical data for the machine to understand.
The dependent variables are:

- 'period'
- 'total_dhis2_value_hts'
- 'total_datim_value_hts'
- 'percentage_difference_hts'
- 'total_dhis2_value_pmtct'
- 'total_datim_value_pmtct'

The target variable is 'accept'

3. Creating and training model

We split the data into train and test dataset with a test dataset taking 20% of the whole data. Then creating the model by importing the Decision tree classifier. Fit the X_train and y_train dataset then predict the X_test.

Model Evaluation

The model's training accuracy was 95% which when compared to the y_test dataset, most were a match. To check if our model was overfitting, the metric used was the F1 score which was 0.98. This means that the model was good.

Model Deployment

The following are a number of reasons why we decided to use Microsoft Azure for deployment of our model:

- scalable - scale up or scale down of the model proved to be really simple with Azure.
- Azure provides very useful tools and platforms which are very useful for machine learning workflows.
- Security - Azure provides a rnage sepreme tools for securing machine learning models which is very valuable for our case.

The following is the process we used to deploy our machine learning:

- Prepared the model - this involved saving the model into a pickel file so that it can be loaded and executed by a web service.
- Created an azure virtual machine learning workspace - this was to be used to store the model and all related files.
- Created a scoring script - this was needed to run and load our model, process input data to generate predictions.
- Built a docker image that included our model, scoring script and any other necessary dependencies.
- Deployed the model - we deployed the docker image to the deployment target.
- We then tested the deployment to ensure it was working correctly.

Challenges

We faced some challenges while working on our solution of building a machine learning solution to solve the problem of inconsistent reporting of HIV testing indicators across facilities. Below are some of the challenges we faced :

1. Unbalanced distribution of PMTCT sites: It's likely that there are far less PMTCT sites than there are sites that report testing. This might lead to a class imbalance issue, in which case our machine learning model might be biased towards the majority class and perform poorly while looking for sites that don't disclose tests.
2. Generalizability: A machine learning model may or may not generalize well to fresh, untried data, even if it performs well on our training data. To make sure that our model is not overfitting to our training data, we must carefully assess its performance on a holdout set of data.