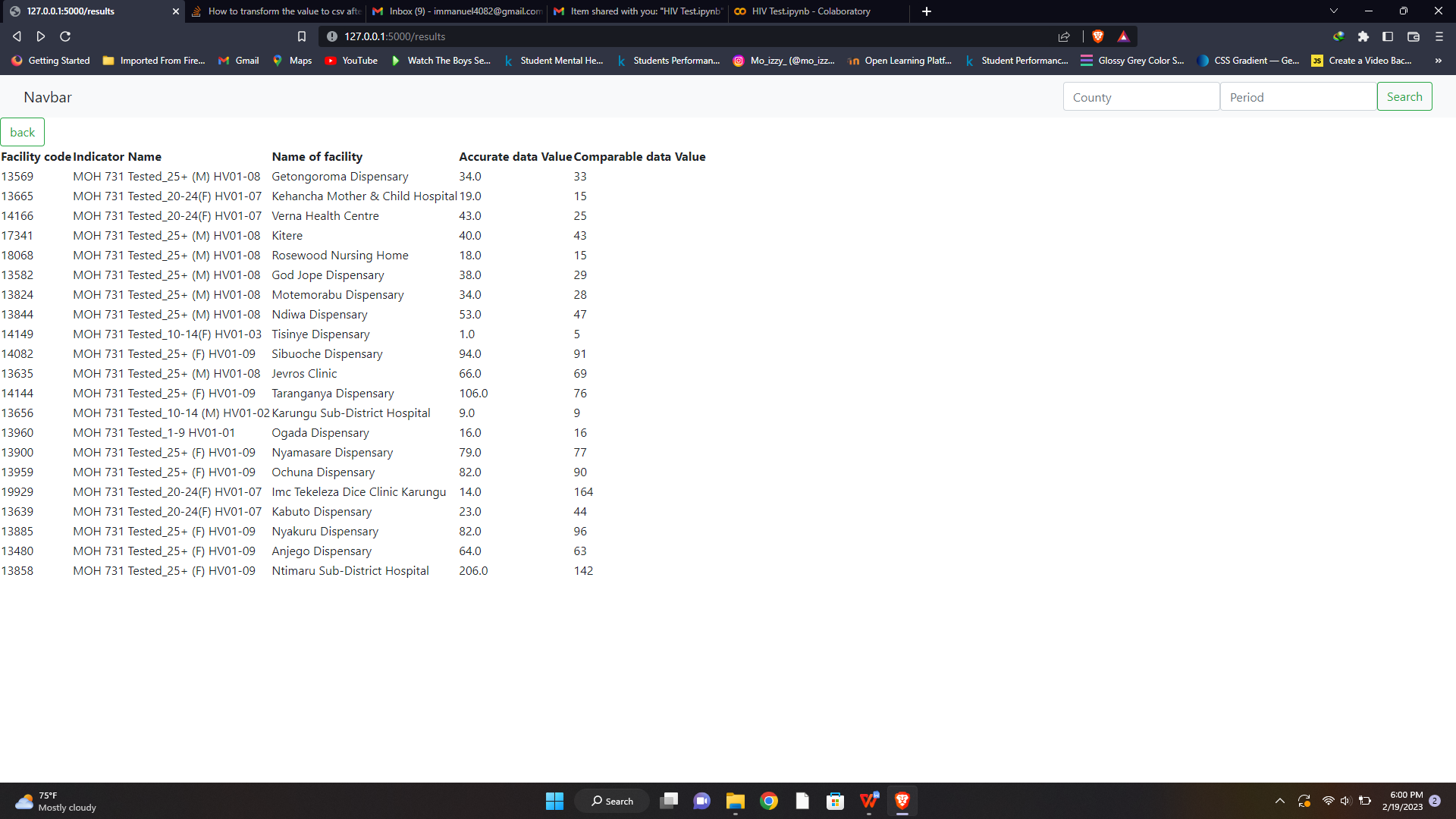
# PMTCT-tests-automation

UI to display Data predicted by the model.



##### Introduction

The HealthIT project continues to support the implementation of various Information Systems on the Ministry of Health. One of the overarching goals of the continued development and deployment of health information systems is to support data driven decision making in the healthcare sector. In this regard a lot of efforts have gone into the development of mechanisms to collect accurate data, to ensure its quality and to develop reports and visualizations.

Our project focused on the development of a classification model app that would check and see which sites did not report their PMTCT - Prevention-Of-Mother-To-Child-Hiv-Transmission. Reprting of HIV testing indicators is inconstistend across facilities some tests include tests from some testing locations and exclude PMTCT tests

#### Objectives of the project

1. To create a model that will predict accurately facilities that have not submitted their PMTCT data
2. Identify other missing variables
3. Be able to flag facilities whose data is inconsistent where both datim\_values and dhis\_values are never aligned
4. To able able to classify and render data in app

##### Value of the data to client

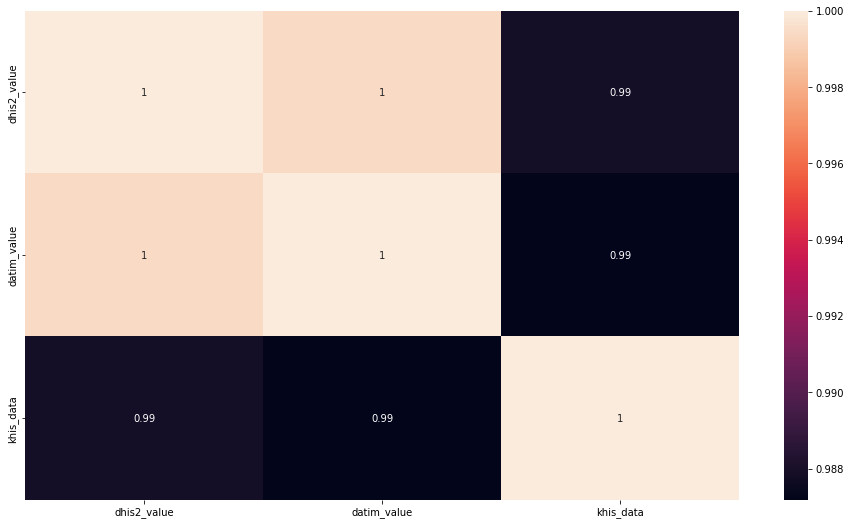
The data for HIV tested in facilities across the countries and the PMTCT data are crucial in both acquisition of drugs and testing kits. Also the amount of HIV positive is important for further policy making. So being able to classify the data and see inconsistency and to be able to flag facilities that are not able to take, store and report the data is important.It will also help them to be able to use data for other various things. We have also created an UI where it will be easy to access the data and to even download a well classified report.

##### Data Acquisition

The data was provided by the Health It team from there systems

###### EDA(Exploratory Data Analysis)

During the EDA process we explored descriptive statistics and multi-coliniarity



Insight -

Most features have a high and direct correlation

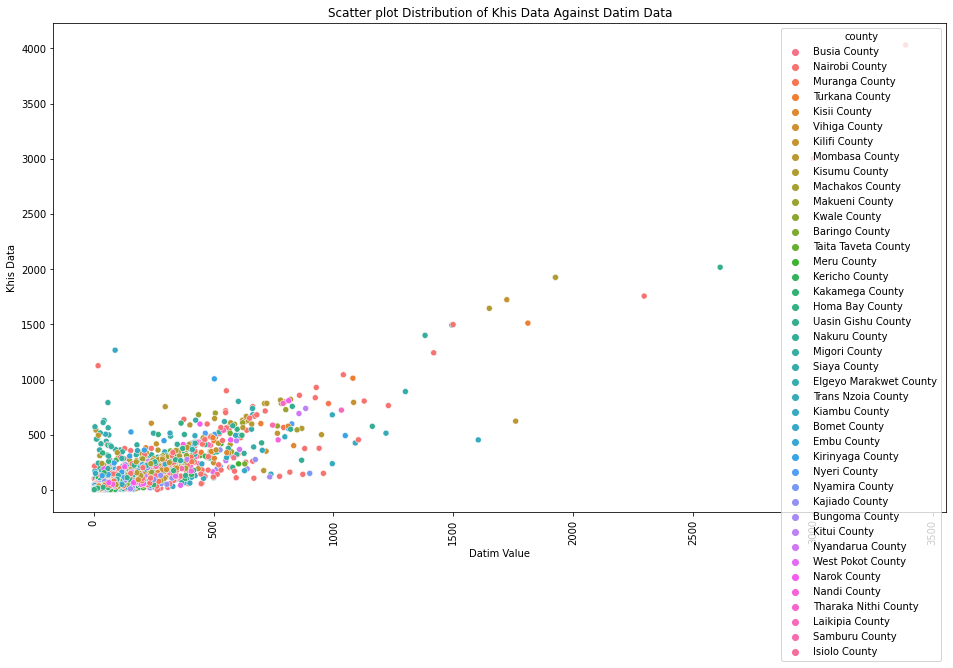
To preprocess the data for a classification model that assists in the identification of PMTCT sites that do not report tests, we can follow these steps:

1. Data Cleaning: We need to check the data for any missing values or errors, and fix or remove them as needed. This ensures that the data is accurate and complete. For instance, we can replace missing values with a mean, median, or mode of the respective feature, or use an interpolation method to fill the missing values.
2. Data Integration: If the data is sourced from different sources or in different formats, we need to combine and convert them into a unified format. For instance, we may need to combine data from different tables, merge or join them based on a common feature, and ensure that the data is in the right format for the analysis.
3. Data Transformation: We may need to transform the data into a more useful format for analysis. For example, we may need to convert categorical data into numerical data using techniques like one-hot encoding or label encoding. We can also transform the data to ensure that it follows a normal distribution or a specific range of values using techniques like logarithmic transformation or min-max scaling.
4. Feature Selection: We need to identify the most important features that contribute significantly to the prediction task. We can use techniques like correlation analysis, feature importance scores, and principal component analysis to select the most relevant features.
5. Data Splitting: Finally, we need to split the data into training and testing sets. The training set is used to train the machine learning model, while the testing set is used to evaluate the performance of the model. We can split the data using techniques like random sampling or stratified sampling to ensure that both the training and testing sets represent the whole dataset.

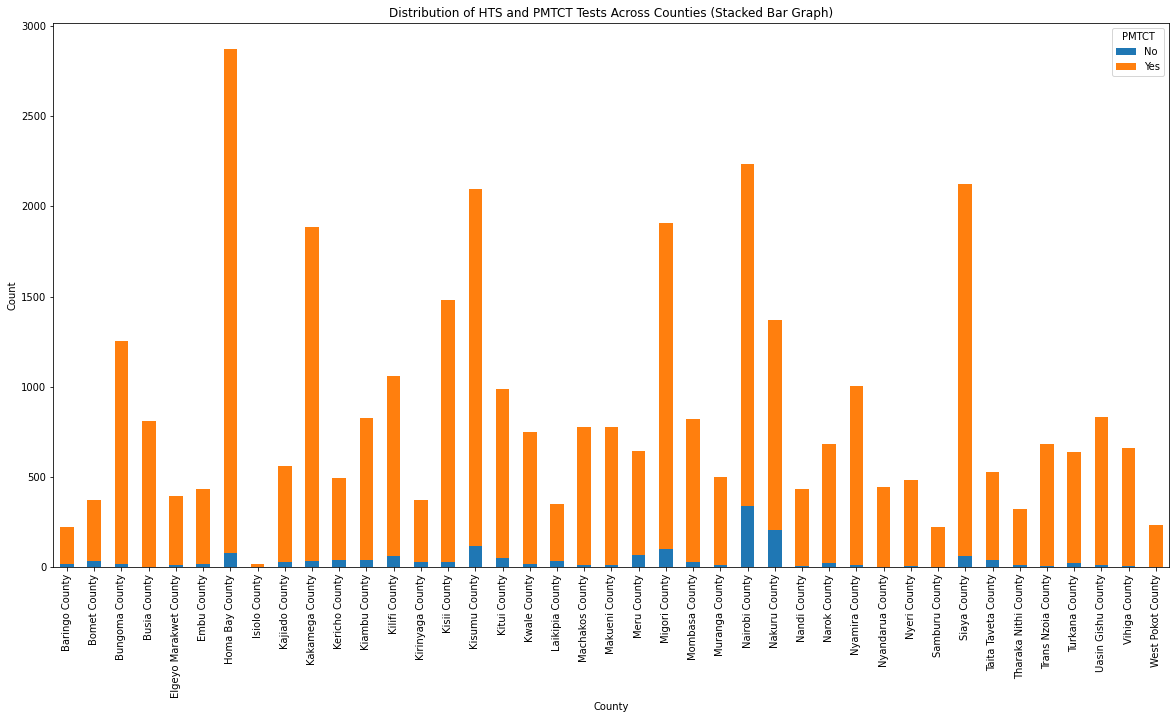
Model creation

To create a classification model to assist in the identification of PMTCT (Prevention of Mother-to-Child Transmission) sites that do not report tests, we need to follow some steps:

1. Collect data: We need to gather data on PMTCT sites that report tests and those that don't. The data should include various features such as the location of the site, the number of patients seen, the number of tests conducted, the type of tests conducted, the number of positive cases, etc.
2. Preprocess data: The data collected will likely be in different formats and may contain missing values. We need to preprocess the data by cleaning it, removing any missing values, and converting it to a format suitable for analysis.
3. Feature selection: We need to identify the most relevant features that can help us identify PMTCT sites that do not report tests. We can use techniques such as correlation analysis and feature importance scores to select the most relevant features.
4. Model selection: There are several machine learning algorithms that we can use to classify PMTCT sites. We need to select an algorithm that can handle our data, is efficient, and has good accuracy. Some popular algorithms for classification include Logistic Regression, Random Forest, Support Vector Machines (SVM), and Neural Networks.



1. Model training: We need to split the data into training and testing sets and use the training set to train the selected machine learning model. We can use techniques such as cross-validation to ensure that the model is not overfitting.



1. Model evaluation: We need to evaluate the performance of the trained model using the testing set. We can use metrics such as accuracy, precision, recall, and F1 score to evaluate the performance of the model.

Precision is a metric that tells us about the quality of positive predictions. Out of every site to have taken PMTCT, how many of them actually did?

Recall tells us about how well the model identifies true positives. Out of all the sites that actually took PMTCT, how many of them did the model identify?

The recall score is the most important metric for this problem. We want to be able to predict as many people who are HIV positive as possible. We don't want to miss out on any of them. This is why we are using the recall score as our main metric for evaluation.

For our case, the recall is 0.93 which is very good. This means that our model is able to predict 93% of the people who are HIV positive correctly.

 Build a Random Forest Classifier

# It returns an error "could not convert string to float" so we need to encode the data

# We will use the OneHotEncoder to encode the data without uisng a pipeline

# We will then use the Random Forest Classifier to fit the model

# Encode the data

encoder = OneHotEncoder(handle\_unknown = "ignore")

X\_train\_encoded = encoder.fit\_transform(X\_train)

X\_test\_encoded = encoder.transform(X\_test)

# Fit the model

rf\_model = RandomForestClassifier(n\_estimators=100, random\_state=42)

rf\_model.fit(X\_train\_encoded, y\_train)

# Make predictions

rf\_y\_pred = rf\_model.predict(X\_test\_encoded)

print(rf\_y\_pred)

# Evaluate the model

from sklearn.metrics import log\_loss, accuracy\_score, classification\_report, roc\_auc\_score, \

confusion\_matrix, ConfusionMatrixDisplay

print(f'Accuracy score: {accuracy\_score(y\_test, y\_pred)}')

print('\n')

print(f'{classification\_report(y\_test, y\_pred)}')

## Output:

['Yes' 'Yes' 'Yes' ... 'Yes' 'Yes' 'Yes'] Accuracy score: 0.9915552427867699 precision recall f1-score support No 1.00 0.80 0.89 302 Yes 0.99 1.00 1.00 6803 accuracy 0.99 7105 macro avg 0.99 0.90 0.94 7105 weighted avg 0.99 0.99 0.99 7105

1. Model deployment: Once we are satisfied with the performance of the model, we can deploy it to assist in the identification of PMTCT sites that do not report tests.

Overall, the creation of a classification model to assist in the identification of PMTCT sites that do not report tests is a complex process that requires careful consideration of the data, feature selection, model selection, training, evaluation, and deployment.

##### Visualizations

