**Prediction of Terrorism incidents in WEST AFRICA**

**Model Refinement**

**1. Overview**

The model refinement phase is a critical step in the machine learning pipeline, focused on improving the performance of the selected model. In this project, I initially experimented with Logistic Regression, Random Forest, Decision Tree, Gradient Boosting, and XGBoost, ultimately choosing XGBoost due to its higher accuracy score.

The prediction result are binary:

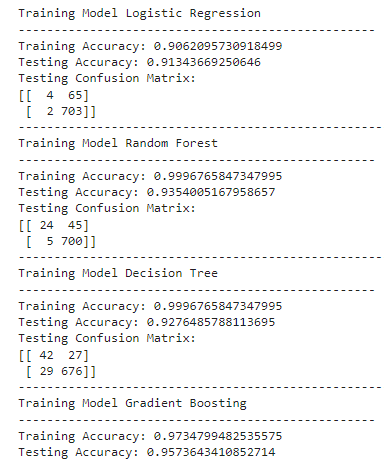
1 for succeeded terrorism attack.

0 for not succeeded terrorism attack.

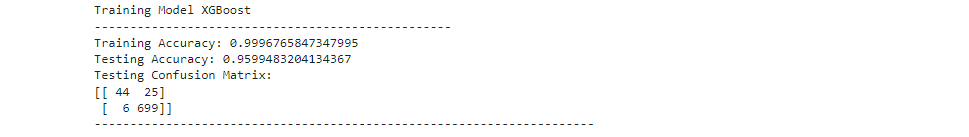
**2. Model Evaluation**

All the above models have been evaluated by using key metrics such as: accuracy, precision, recall and F1.

Their accuracy on training and testing set have been observed.



For the predictions the best model chosen is XGB due to his shown high performance on both train and test set.

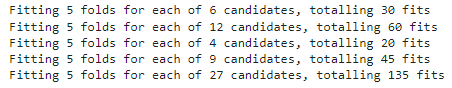


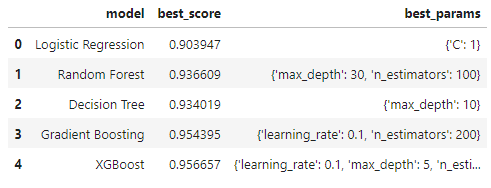
**3. Refinement Techniques**

In this section, Hyperparameter technique with cross-validation have been chosen to refine the model. Refinement techniques have been applied to all the models in order to find better parameters for each model.

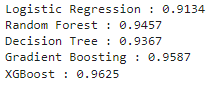
**4. Hyperparameter Tuning and Cross Validation**

During the refinement of those models GridSearchCV from Scite-learn library has been used. The number of cross-validation folds has been set to 5 and as an appropriate metric accuracy has been chosen over precision and recall.



The best scores and best parameters for each model obtained after hyperparameter tuning and cross-validation are shown in the image below:  


After fındıng the best parameters and best score for each model, the test set has been used to evaluated the accuracy for each model and the result found for each model are shown below:



**6. Feature Selection**

First of all, initially we had a huge dataset with more 180000 records and number of feature of 135. After specifying the prediction to be only in West Africa area we got 5136 entries.

The dataset contains features or columns with both categorical and numerical values. Analysis has been done and each feature containing more than 25% of missing values has been dropped.

We finally found our data with 48 features.

**Test Submission**

**1. Overview**

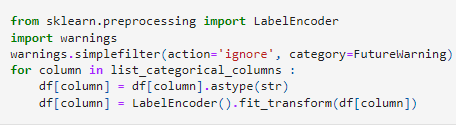
After choosing the best model for the prediction of terrorism attack. User defined data has been used as input data to test the XGB model. Before that a sample of our collected data set has been used to test the model

**2. Data Preparation for Testing**

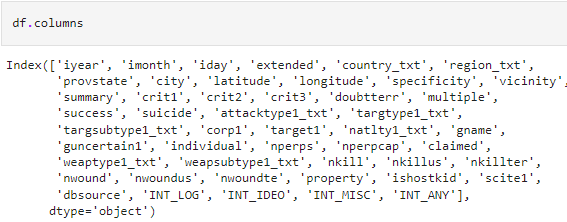
When preparing data for testing, specific and useful columns have been chosen and the ones that are not useful have been dropped.



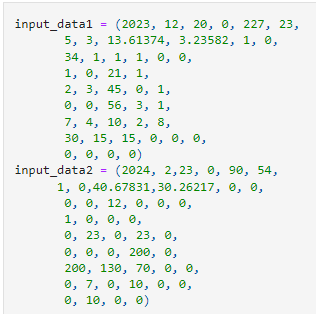
After that Encoding has been proceeded for categorical features.



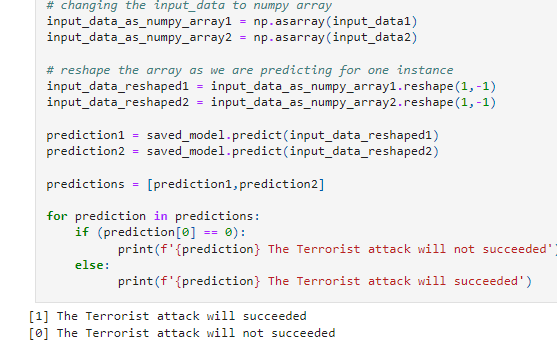
The final chosen columns for testing are shown below,and based on this columns we create our own data as inputs to test the model also.



Data examples defined as inputs for test.



The result for this two data inputs:



**3. Model Application**

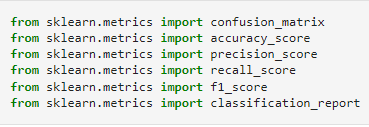
The model has been save by using the pickle library. Pickle is a Python module that provides a way to serialize and deserialize Python objects. Serialization is the process of converting a Python object into a byte stream, and deserialization is the process of reconstructing the original object from the byte stream.

After that the model has been loaded again for test.

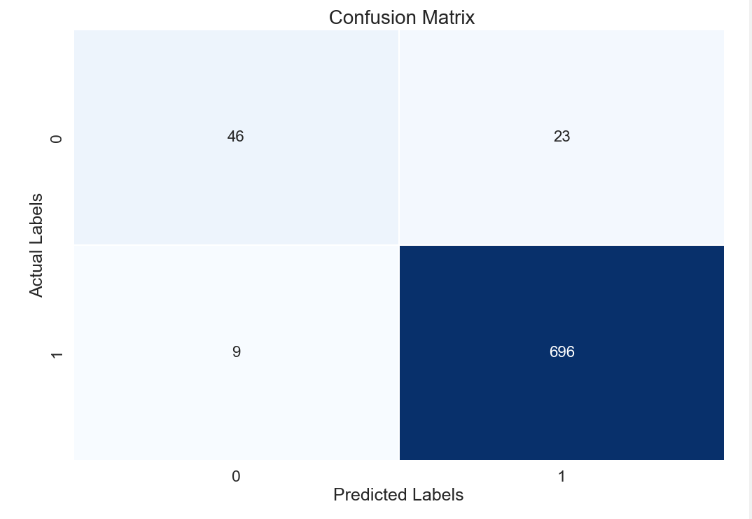


**4. Test Metrics**

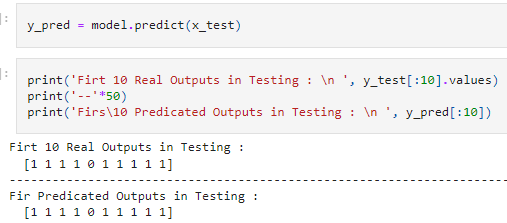
Several metrics have been used to evaluate the model's performance on the dataset which has been splited in 80% of training and 20% of testing. The metrics used for evaluation are shown below:

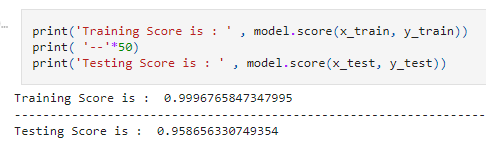


The confusion matrix of the chosen model is shown below:

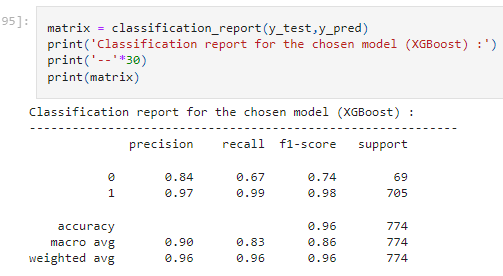


10 First predictions when testing made and the accuracy of XGB on both Training and Testing:





The model’s report:



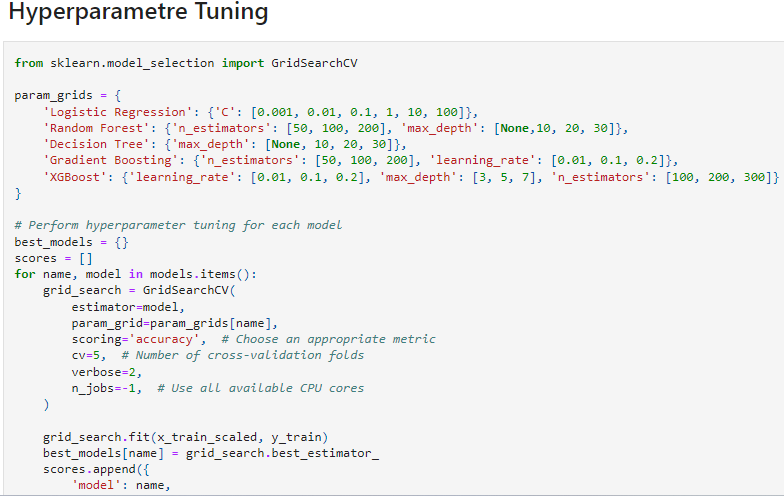
**5. Model Deployment**

For the deployment of the model, I plan to use the most popular user interface library for complicated systems React. React is commonly used for developing real-world web application that will be in production (will be used in real life scenario).

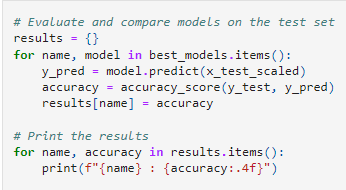
An entire Full Stack (Frontend and Backend) web application will be built by integrating inside our prediction model. Finally, a warning system will be integrated in the application in case the terrorism attack is more likely to happen.

**6. Code Implementation**

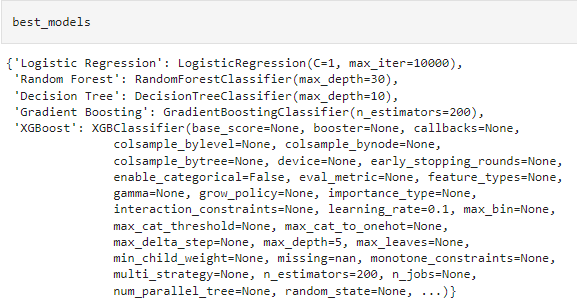
After choosing the right models for the project and after training those models. We get into Hyperparameter tuning in order to increase the performance the chosen model.



The code below compares and evaluates those models on the test set.



The final best models including best parameters.



**Conclusion**

Ultimately, the model refinement plays an important rule concerning the model performance improvement. After applying Hyperparameter methods the model chosen predict an eventual terrorist attack with really height accuracy with is more than 95%.

The difficulties I may encounter is mismatches between JavaScript Framework React and our python based model.

**References**

HUAMANÍ, Enrique Lee, ALVA, Mantari Alicia, et ROMAN-GONZALEZ, Avid. Machine learning techniques to visualize and predict terrorist attacks worldwide using the global terrorism database. International Journal of Advanced Computer Science and Applications, 2020, vol. 11, no 4.

DEMIR, Mustafa, GULER, Ahmet, et OZER, Murat. Predictors of successful terrorism incidents. Behavioral Sciences of Terrorism and Political Aggression, 2022, p. 1-29..

https://www.wearecapicua.com/blog/machine-learning-reactjs