**Cyber Breach Detection**

# Architecture

# Root Directories

## Directories

|  |  |  |
| --- | --- | --- |
| Directories | Files under Directories | Dir Explanation |
| Common | Pipeline  Constant\_variables  Custom\_encoder  Enums  Utils\_external | Mostly contains files that are used by training as well prediction modules |
| Config | Config.yaml | Configuration file |
| Datasets | 0 / 1 | Dataset with label folder |
| Flow | Flowchart.drawio | DRAFT Flow diagrams |
| Models | Xgb\_model2.pkl | Trained models |
| Notes | N/A | How to execute and documentation files |
| Predicted\_reports | N/A | Predicted reports |
| prediction | Prediction\_repositories | Contains all methods regarding prediction |
| Test | Test files | Contains test files to test entire program |
| training | Training\_repositories | Contains all methods regarding training |

## Root Files

|  |  |
| --- | --- |
| Directories | Files under Directories |
| .gitignore | Ignores files which are not required to be stored in git repo |
| App.py | Separates different executions - Training or Predicting |
| Main.py | Executes program through commandline – Executes app |
| Submit\_test.py | Execution starts here |
| Requirements.txt | Includes all libraries required to run the application |

# Files under each Root Dir

## Common

* Pipeline
  + Utils.py – Contains methods for creating entire pipeline for prediction and training. Reads data, cleans etc.
* Constant\_variables – contains file that stores features in constant variables
* Enums.py – Contains constant enums for assigning proper datatype for each type of application
* Utils\_external.py – In simple words, utility package. Contains methods that may be used by multiple applications or files.s

## Config

* Config.yaml – This file stores configuration of the application.

## Datasets

* 0 – Should contain dataset that will be labelled as 0 – Non Breached data.
* 1 – Should contain dataset that will be labelled as 1 – Breached data.

## Models

* Xgb\_model2.pkl – Model file that is used to predict the probability of website getting breached.

## Notes

* Documentation – This document
* HOW TO RUN – How to run this program

## Predicted\_reports

* This folder stores all the predicted outputs. First we need to train the model and then predict.

## Prediction

* Prediction \_repository - This repo contains all the methods regarding making predictions.

## Training

* Training\_repository – This repo contains all the methods regarding making training.

## App.py

* This module is used to run the entire application.
* This module helps to separate training and prediction modules.

## Requirements.txt

* It lists out all the library dependencies used by the application.

## Submit\_test.py

* This file helps to test without the use of external sources. Helps to simulate production environment.