Import Libraries

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn
import shap

import xgboost as xgb
from xgboost import XGBClassifier, XGBRegressor
from xgboost import to_graphviz, plot_importance

%matplotlib inline
sns.set_style('dark')
sns.set(font_scale=1.2)

from sklearn.inspection import permutation_importance
from sklearn.model_selection import cross_val_score, train_test_split, GridSearchCV, RandomizedSearchCV
from sklearn.preprocessing import LabelEncoder, StandardScaler, MinMaxScaler, OneHotEncoder
from sklearn.pipeline import Pipeline
from sklearn.metrics import confusion_matrix, classification_report, mean_absolute_error, mean_squared_
error,r2_score
from sklearn.metrics import plot_confusion_matrix, plot_precision_recall_curve, plot_roc_curve, accurac
y_score
from sklearn.metrics import auc, f1_score, precision_score, recall_score, roc_auc_score


import warnings
warnings.filterwarnings('ignore')

import pickle
from pickle import dump, load

np.random.seed(0)

#from pycaret.classification import *
#from pycaret.clustering import *
#from pycaret.regression import *

pd.set_option('display.max_columns',100)
#pd.set_option('display.max_rows',100)
pd.set_option('display.width', 1000)
```

Load the Model

In [2]:

```
filename = 'pumpmodel.sav'
```

In [3]:

```
loaded_model = load(open(filename,'rb'))
```

In [4]:

```
loaded_model
```

Out[4]:

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=0.8, eta=0.15, gamma=4,
              gpu_id=-1, importance_type='gain', interaction_constraints='',
              learning_rate=0.150000006, max_delta_step=0, max_depth=9,
              min_child_weight=3, missing=nan, monotone_constraints='()',
              n_estimators=600, n_jobs=0, num_parallel_tree=1, random_state=0,
              reg_alpha=0.0, reg_lambda=0.9, scale_pos_weight=1, subsample=0.8,
              tree_method='exact', validate_parameters=1, verbosity=None)
```

In [5]:

```
test = pd.read_csv("test5.csv")
```

In [6]:

```
test
```

Out[6]:

	amount_tsh	age	population	basin_Lake Nyasa	basin_Lake Rukwa	basin_Lake Tanganyika	basin_Lake Victoria	basin_Pangani	basin_Rufiji	basin_Ruvuma / Southern Coast	basin_ /
0	0.0	8	321	0	0	0	0	0	0	0	
1	0.0	20	300	0	0	0	0	1	0	0	
2	0.0	30	200	0	0	0	0	1	0	0	
3	0.0	13	600	0	0	0	0	0	1	0	
4	0.0	38	1	0	0	0	0	0	0	1	
...	...	...	...	...	...	...	...	...	...	...	
8068	0.0	25	1140	0	0	1	0	0	0	0	
8069	0.0	32	20	0	0	0	0	0	0	0	
8070	1000.0	26	1140	0	0	0	0	1	0	0	
8071	0.0	11	150	1	0	0	0	0	0	0	
8072	0.0	12	40	1	0	0	0	0	0	0	

8073 rows × 56 columns

In [7]:

```
y_pred = loaded_model.predict(test)
```

In [8]:

```
y_pred
```

Out[8]:

```
array([1, 0, 0, ..., 0, 0, 1], dtype=int64)
```

In [9]:

```
prediction = pd.DataFrame(y_pred, columns=["Pump Prediction"])
```

In [10]:

```
prediction
```

Out[10]:

	Pump Prediction
0	1
1	0
2	0
3	0
4	1
...	...
8068	1
8069	1
8070	0
8071	0
8072	1

8073 rows × 1 columns

In [ ]: