PACKAGES AND LIBRARIES

add Codeadd Markdown

#GENERAL

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

#PATH PROCESS

import os

import os.path

from pathlib import Path

import glob

#IMAGE PROCESS

from PIL import Image

from keras.preprocessing import image

from tensorflow.keras.preprocessing.image import ImageDataGenerator

import cv2

from keras.applications.vgg16 import preprocess\_input, decode\_predictions

#SCALER & TRANSFORMATION

from sklearn.preprocessing import StandardScaler

from sklearn.preprocessing import MinMaxScaler

from keras.utils.np\_utils import to\_categorical

from sklearn.model\_selection import train\_test\_split

from keras import regularizers

from sklearn.preprocessing import LabelEncoder

#ACCURACY CONTROL

from sklearn.metrics import confusion\_matrix, accuracy\_score, classification\_report, roc\_auc\_score, roc\_curve

from sklearn.model\_selection import GridSearchCV, cross\_val\_score

from sklearn.metrics import mean\_squared\_error, r2\_score

#OPTIMIZER

from keras.optimizers import RMSprop,Adam,Optimizer,Optimizer

#MODEL LAYERS

from tensorflow.keras.models import Sequential

from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D, BatchNormalization,MaxPooling2D,BatchNormalization,\

Permute, TimeDistributed, Bidirectional,GRU, SimpleRNN, LSTM, GlobalAveragePooling2D, SeparableConv2D

from keras import models

from keras import layers

import tensorflow as tf

from keras.applications import VGG16,VGG19,inception\_v3

from keras import backend as K

from keras.utils import plot\_model

#SKLEARN CLASSIFIER

from xgboost import XGBClassifier, XGBRegressor

from lightgbm import LGBMClassifier, LGBMRegressor

from catboost import CatBoostClassifier, CatBoostRegressor

from sklearn.linear\_model import LogisticRegression

from sklearn.naive\_bayes import GaussianNB

from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor

from sklearn.ensemble import GradientBoostingClassifier, GradientBoostingRegressor

from sklearn.ensemble import BaggingRegressor

from sklearn.tree import DecisionTreeClassifier, DecisionTreeRegressor

from sklearn.neural\_network import MLPClassifier, MLPRegressor

from sklearn.neighbors import KNeighborsClassifier, KNeighborsRegressor

from sklearn.linear\_model import LinearRegression

from sklearn.cross\_decomposition import PLSRegression

from sklearn.linear\_model import Ridge

from sklearn.linear\_model import RidgeCV

from sklearn.linear\_model import Lasso

from sklearn.linear\_model import LassoCV

from sklearn.linear\_model import ElasticNet

from sklearn.linear\_model import ElasticNetCV

#IGNORING WARNINGS

from warnings import filterwarnings

filterwarnings("ignore",category=DeprecationWarning)

filterwarnings("ignore", category=FutureWarning)

filterwarnings("ignore", category=UserWarning)

add Codeadd Markdown

PATH & LABEL PROCESS

add Codeadd Markdown

MAIN PATH

add Codeadd Markdown

Fire\_Dataset\_Path = Path("../input/fire-dataset/fire\_dataset")

add Codeadd Markdown

PATH PROCESS

add Codeadd Markdown

PNG\_Path = list(Fire\_Dataset\_Path.glob(r"\*/\*.png"))

add Codeadd Markdown

LABEL PROCESS

add Codeadd Markdown

PNG\_Labels = list(map(lambda x: os.path.split(os.path.split(x)[0])[1],PNG\_Path))

add Codeadd Markdown

print("FIRE: ", PNG\_Labels.count("fire\_images"))

print("NO\_FIRE: ", PNG\_Labels.count("non\_fire\_images"))