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"))
```