PROCEDURE TO WORK KAGGLE FOR EARTH QUAKE PREDICTION MODEL USING PYTHON

DATA BASE CSV

```
IMPORT NUMPY AS NP
IMPORT PANDAS AS PD
IMPORT MATPLOTLIB.PYPLOT AS PLT

IMPORT OS
PRINT(OS.LISTDIR("../INPUT"))
```

```
['DATABASE.CSV']
```

READ THE DATA FROM CSV AND ALSO COLUMNS WHICH ARE NECESSARY FOR THE MODEL AND THE COLUMN WHICH NEEDS TO BE PREDICTED.

IN [2]:

```
DATA = PD.READ_CSV("../INPUT/DATABASE.CSV")
DATA.HEAD()
```

IN [3]:

```
DATA.COLUMNS
```

```
INDEX(['DATE', 'TIME', 'LATITUDE', 'LONGITUDE', 'TYPE', 'DEPTH', 'DEPTH ERROR',
       'DEPTH SEISMIC STATIONS', 'MAGNITUDE', 'MAGNITUDE TYPE',
       'MAGNITUDE ERROR', 'MAGNITUDE SEISMIC STATIONS', 'AZIMUTHAL GAP',
       'HORIZONTAL DISTANCE', 'HORIZONTAL ERROR', 'ROOT MEAN SQUARE', 'ID',
       'SOURCE', 'LOCATION SOURCE', 'MAGNITUDE SOURCE', 'STATUS'],
      DTYPE='OBJECT')
```

FIGURE OUT THE MAIN FEATURES FROM EARTHQUAKE DATA AND CREATE A OBJECT OF THAT FEATURES, NAMELY, DATE, TIME, LATITUDE, LONGITUDE, DEPTH, MAGNITUDE.

IN [4]:

```
DATA = DATA[['DATE', 'TIME', 'LATITUDE', 'LONGITUDE', 'DEPTH', 'MAGNITUDE']]
DATA.HEAD()
```

OUT[4]:

	DATE	TIME	LATITUDE	LONGITUDE	DEPTH	MAGNITUDE
0	01/02/1965	13:44:18	19.246	145.616	131.6	6.0
1	01/04/1965	11:29:49	1.863	127.352	80.0	5.8
2	01/05/1965	18:05:58	-20.579	-173.972	20.0	6.2
3	01/08/1965	18:49:43	-59.076	-23.557	15.0	5.8
4	01/09/1965	13:32:50	11.938	126.427	15.0	5.8

HERE, THE DATA IS RANDOM WE NEED TO SCALE ACCORDING TO INPUTS TO THE MODEL. IN THIS, WE CONVERT GIVEN DATE AND TIME TO UNIX TIME WHICH IS IN SECONDS AND A NUMERAL. THIS CAN BE EASILY USED AS INPUT FOR THE NETWORK WE BUILT.

IN [5]:

```
IMPORT DATETIME
IMPORT TIME

TIMESTAMP = []
FOR D, T IN ZIP(DATA['DATE'], DATA['TIME']):
    TRY:
        TS = DATETIME.DATETIME.STRPTIME(D+' '+T, '%M/%D/%Y %H:%M:%S')
        TIMESTAMP.APPEND(TIME.MKTIME(TS.TIMETUPLE()))
    EXCEPT VALUEERROR:
        # PRINT('VALUEERROR')
        TIMESTAMP.APPEND('VALUEERROR')
```

IN [6]:

```
TIMESTAMP = PD.SERIES(TIMESTAMP)
DATA['TIMESTAMP'] = TIMESTAMP.VALUES
```

IN [7]:

```
FINAL_DATA = DATA.DROP(['DATE', 'TIME'], AXIS=1)
FINAL_DATA = FINAL_DATA[FINAL_DATA.TIMESTAMP != 'VALUEERROR']
```

FINAL_DATA.HEAD()

OUT[7]:

	LATITUDE	LONGITUDE	DEPTH	MAGNITUDE	TIMESTAMP
0	19.246	145.616	131.6	6.0	-1.57631E+08
1	1.863	127.352	80.0	5.8	-1.57466E+08
2	-20.579	-173.972	20.0	6.2	-1.57356E+08
3	-59.076	-23.557	15.0	5.8	-1.57094E+08
4	11.938	126.427	15.0	5.8	-1.57026E+08

VISUALIZATION

HERE, ALL THE EARTHQUAKES FROM THE DATABASE IN VISUALIZED ON TO THE WORLD MAP WHICH SHOWS CLEAR REPRESENTATION OF THE LOCATIONS WHERE FREQUENCY OF THE EARTHQUAKE WILL BE MORE.

IN [8]:

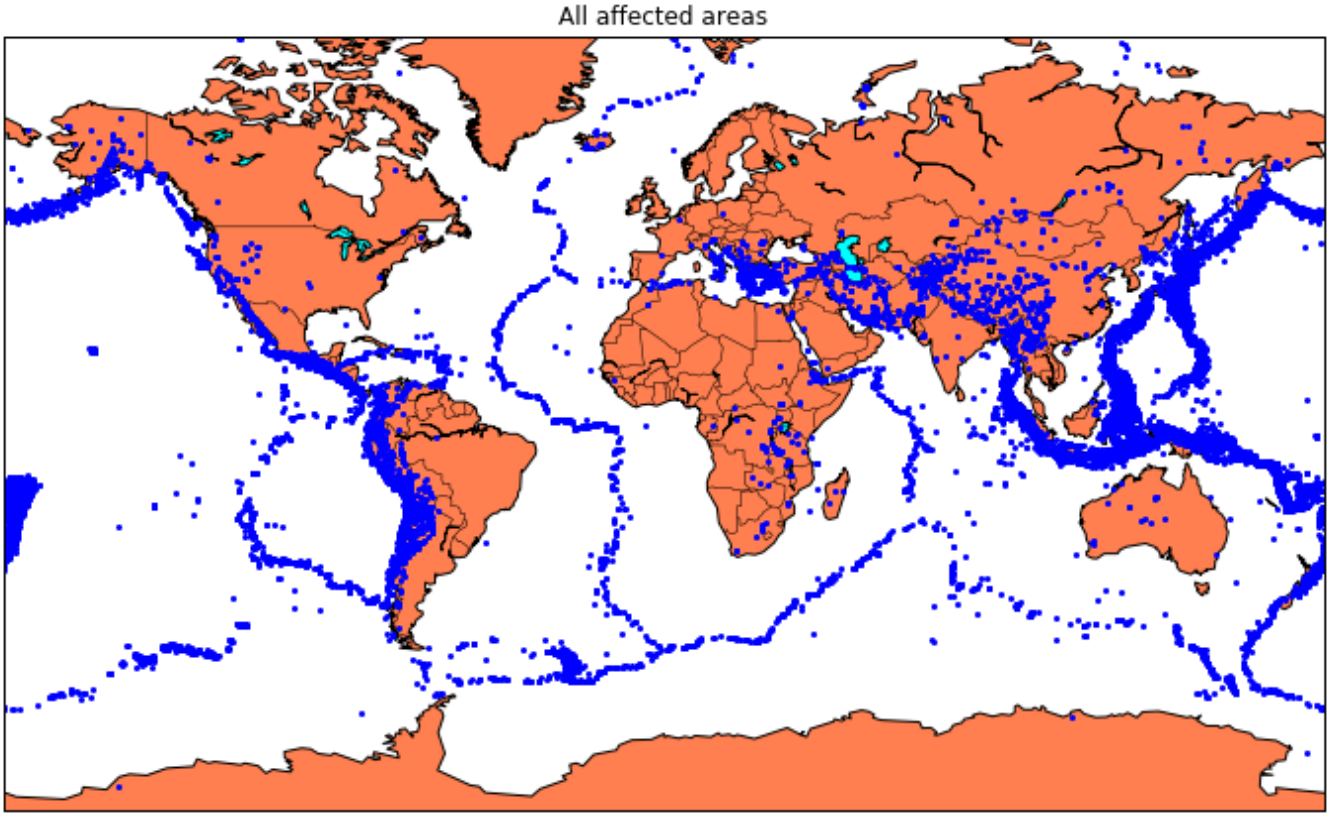
```
FROM MPL_TOOLKITS.BASEMAP IMPORT BASEMAP

M = BASEMAP(PROJECTION='MILL',LLCRNRLAT=-80,URCRNRLAT=80, LLCRNRLON
=-180,URCRNRLON=180,LAT_TS=20,RESOLUTION='C')

LONGITUDES = DATA["LONGITUDE"].TOLIST()
LATITUDES = DATA["LATITUDE"].TOLIST()
#M = BASEMAP(WIDTH=12000000,HEIGHT=9000000,PROJECTION='LCC',
        #RESOLUTION=NONE,LAT_1=80.,LAT_2=55,LAT_0=80,LON_0=-107.)
X,Y = M(LONGITUDES,LATITUDES)
```

IN [9]:

```
FIG = PLT.FIGURE(FIGSIZE=(12,10))
PLT.TITLE("ALL AFFECTED AREAS")
M.PLOT(X, Y, "O", MARKERSIZE = 2, COLOR = 'BLUE')
M.DRAWCOASTLINES()
```

```
M.FILLCONTINENTS(COLOR='CORAL',LAKE_COLOR='AQUA')
M.DRAWMAPBOUNDARY()
M.DRAWCOUNTRIES()
PLT.SHOW()
```

```
/OPT/CONDA/LIB/PYTHON3.6/SITE-PACKAGES/MPL_TOOLKITS/BASEMAP/__INIT
__.PY:1704: MATPLOTLIBDEPRECATIONWARNING: THE AXESPATCH FUNCTION WA
S DEPRECATED IN VERSION 2.1. USE AXES.PATCH INSTEAD.
  LIMB = AX.AXESPATCH
/OPT/CONDA/LIB/PYTHON3.6/SITE-PACKAGES/MPL_TOOLKITS/BASEMAP/__INIT
__.PY:1707: MATPLOTLIBDEPRECATIONWARNING: THE AXESPATCH FUNCTION WA
S DEPRECATED IN VERSION 2.1. USE AXES.PATCH INSTEAD.
  IF LIMB IS NOT AX.AXESPATCH:
```



SPLITTING THE DATA

FIRSTLY, SPLIT THE DATA INTO XS AND YS WHICH ARE INPUT TO THE MODEL AND OUTPUT OF THE MODEL RESPECTIVELY. HERE, INPUTS ARE TIMESTAMP, LATITUDE AND LONGITUDE AND OUTPUTS ARE MAGNITUDE AND DEPTH. SPLIT THE XS AND YS INTO TRAIN AND TEST WITH VALIDATION. TRAINING DATASET CONTAINS 80% AND TEST DATASET CONTAINS 20%.

IN [10]:

```
X = FINAL_DATA[['TIMESTAMP', 'LATITUDE', 'LONGITUDE']]
Y = FINAL_DATA[['MAGNITUDE', 'DEPTH']]
```

IN [11]:

```
FROM SKLEARN.CROSS_VALIDATION IMPORT TRAIN_TEST_SPLIT
```

```
X_TRAIN, X_TEST, Y_TRAIN, Y_TEST = TRAIN_TEST_SPLIT(X, Y, TEST_SIZE=0.2,
RANDOM_STATE=42)
PRINT(X_TRAIN.SHAPE, X_TEST.SHAPE, Y_TRAIN.SHAPE, X_TEST.SHAPE)
```

```
(18727, 3) (4682, 3) (18727, 2) (4682, 3)
/OPT/CONDA/LIB/PYTHON3.6/SITE-PACKAGES/SKLEARN/CROSS_VALIDATION.P
Y:41: DEPRECATIONWARNING: THIS MODULE WAS DEPRECATED IN VERSION 0.18 I
N FAVOR OF THE MODEL_SELECTION MODULE INTO WHICH ALL THE REFACTORED
CLASSES AND FUNCTIONS ARE MOVED. ALSO NOTE THAT THE INTERFACE OF THE
NEW CV ITERATORS ARE DIFFERENT FROM THAT OF THIS MODULE. THIS MODULE
WILL BE REMOVED IN 0.20.
  "THIS MODULE WILL BE REMOVED IN 0.20.", DEPRECATIONWARNING)
```

HERE, WE USED THE RANDOMFORESTREGRESSOR MODEL TO PREDICT THE OUTPUTS, WE SEE THE STRANGE PREDICTION FROM THIS WITH SCORE ABOVE 80% WHICH CAN BE ASSUMED TO BE BEST FIT BUT NOT DUE TO ITS PREDICTED VALUES.

IN [12]:

```
FROM SKLEARN.ENSEMBLE IMPORT RANDOMFORESTREGRESSOR

REG = RANDOMFORESTREGRESSOR(RANDOM_STATE=42)
REG.FIT(X_TRAIN, Y_TRAIN)
REG.PREDICT(X_TEST)
```

```
/OPT/CONDA/LIB/PYTHON3.6/SITE-PACKAGES/SKLEARN/ENSEMBLE/WEIGHT_B
OOSTING.PY:29: DEPRECATIONWARNING: NUMPY.CORE.UMATH_TESTS IS AN INTE
RNAL NUMPY MODULE AND SHOULD NOT BE IMPORTED. IT WILL BE REMOVED IN A
FUTURE NUMPY RELEASE.
  FROM NUMPY.CORE.UMATH_TESTS IMPORT INNER1D

ARRAY([[  5.96,  50.97],
       [  5.88,  37.8 ],
       [  5.97,  37.6 ],
       ...,
       [  6.42,  19.9 ],
       [  5.73, 591.55],
       [  5.68,  33.61]])
```

IN [13]:

```
REG.SCORE(X_TEST, Y_TEST)
```

IN [14]:

```
FROM SKLEARN.MODEL_SELECTION IMPORT GRIDSEARCHCV
```

```
PARAMETERS = {'N_ESTIMATORS':[10, 20, 50, 100, 200, 500]}

GRID_OBJ = GRIDSEARCHCV(REG, PARAMETERS)
GRID_FIT = GRID_OBJ.FIT(X_TRAIN, Y_TRAIN)
BEST_FIT = GRID_FIT.BEST_ESTIMATOR_
BEST_FIT.PREDICT(X_TEST)
```

IN [15]:

```
BEST_FIT.SCORE(X_TEST, Y_TEST)
```