Main.py

from tkinter import messagebox

from tkinter import \*

from tkinter import simpledialog

import tkinter

from tkinter import filedialog

import matplotlib.pyplot as plt

import numpy as np

from tkinter.filedialog import askopenfilename

import pandas as pd

from sklearn import \*

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

from keras.models import Sequential

from keras.layers.core import Dense,Activation,Dropout

from keras.callbacks import EarlyStopping

from sklearn.preprocessing import OneHotEncoder

from keras.optimizers import Adam

import os

main = tkinter.Tk()

main.title("Average Fuel Consumption") #designing main screen

main.geometry("1300x1200")

global filename

global train\_x, test\_x, train\_y, test\_y

global balance\_data

global model

global ann\_acc

global testdata

global predictdata

def importdata():

global balance\_data

balance\_data = pd.read\_csv(filename)

balance\_data = balance\_data.abs()

return balance\_data

def splitdataset(balance\_data):

global train\_x, test\_x, train\_y, test\_y

X = balance\_data.values[:, 0:7]

y\_ = balance\_data.values[:, 7]

print(y\_)

y\_ = y\_.reshape(-1, 1)

encoder = OneHotEncoder(sparse=False)

Y = encoder.fit\_transform(y\_)

print(Y)

train\_x, test\_x, train\_y, test\_y = train\_test\_split(X, Y, test\_size=0.2)

text.insert(END,"Dataset Length : "+str(len(X))+"\n");

return train\_x, test\_x, train\_y, test\_y

def upload(): #function to upload tweeter profile

global filename

filename = filedialog.askopenfilename(initialdir="dataset")

text.delete('1.0', END)

text.insert(END,filename+" loaded\n\n");

def generateModel():

global train\_x, test\_x, train\_y, test\_y

data = importdata()

train\_x, test\_x, train\_y, test\_y = splitdataset(data)

text.insert(END,"Splitted Training Length : "+str(len(train\_x))+"\n");

text.insert(END,"Splitted Test Length : "+str(len(test\_x))+"\n");

def ann():

global model

global ann\_acc

model = Sequential()

model.add(Dense(200, input\_shape=(7,), activation='relu', name='fc1'))

model.add(Dense(200, activation='relu', name='fc2'))

model.add(Dense(19, activation='softmax', name='output'))

optimizer = Adam(lr=0.001)

model.compile(optimizer, loss='categorical\_crossentropy', metrics=['accuracy'])

print('CNN Neural Network Model Summary: ')

print(model.summary())

model.fit(train\_x, train\_y, verbose=2, batch\_size=5, epochs=200)

results = model.evaluate(test\_x, test\_y)

text.insert(END,"ANN Accuracy for dataset "+filename+"\n");

text.insert(END,"Accuracy Score : "+str(results[1]\*100)+"\n\n")

ann\_acc = results[1] \* 100

def predictFuel():

global testdata

global predictdata

text.delete('1.0', END)

filename = filedialog.askopenfilename(initialdir="dataset")

testdata = pd.read\_csv(filename)

testdata = testdata.values[:, 0:7]

predictdata = model.predict\_classes(testdata)

print(predictdata)

for i in range(len(testdata)):

text.insert(END,str(testdata[i])+" Average Fuel Consumption : "+str(predictdata[i])+"\n");

def graph():

x = []

y = []

for i in range(len(testdata)):

x.append(i)

y.append(predictdata[i])

plt.plot(x, y)

plt.xlabel('Vehicle ID')

plt.ylabel('Fuel Consumption/10KM')

plt.title('Average Fuel Consumption Graph')

plt.show()

font = ('times', 16, 'bold')

title = Label(main, text='A Machine Learning Model for Average Fuel Consumption in Heavy Vehicles')

title.config(bg='greenyellow', fg='dodger blue')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=0,y=5)

font1 = ('times', 12, 'bold')

text=Text(main,height=20,width=150)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=50,y=120)

text.config(font=font1)

font1 = ('times', 14, 'bold')

uploadButton = Button(main, text="Upload Heavy Vehicles Fuel Dataset", command=upload)

uploadButton.place(x=50,y=550)

uploadButton.config(font=font1)

modelButton = Button(main, text="Read Dataset & Generate Model", command=generateModel)

modelButton.place(x=420,y=550)

modelButton.config(font=font1)

annButton = Button(main, text="Run ANN Algorithm", command=ann)

annButton.place(x=760,y=550)

annButton.config(font=font1)

predictButton = Button(main, text="Predict Average Fuel Consumption", command=predictFuel)

predictButton.place(x=50,y=600)

predictButton.config(font=font1)

graphButton = Button(main, text="Fuel Consumption Graph", command=graph)

graphButton.place(x=420,y=600)

graphButton.config(font=font1)

exitButton = Button(main, text="Exit", command=exit)

exitButton.place(x=760,y=600)

exitButton.config(font=font1)

main.config(bg='LightSkyBlue')

main.mainloop()