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 numpy as np

import pandas as pd

from sklearn import \*

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

from sklearn.metrics import accuracy\_score

from sklearn.metrics import classification\_report

from sklearn.ensemble import RandomForestClassifier

#from sklearn.tree import export\_graphviz

#from IPython import display

main = tkinter.Tk()

main.title("Fraud Detection System Using ML") #designing main screen

main.geometry("1300x1200")

global filename

global cls

global X, Y, X\_train, X\_test, y\_train, y\_test

global random\_acc # all global variables names define in above lines

global clean

global attack

global total

def traintest(train): #method to generate test and train data from dataset

X = train.values[:, 0:29]

Y = train.values[:, 30]

print(X)

print(Y)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

X, Y, test\_size = 0.3, random\_state = 0)

return X, Y, X\_train, X\_test, y\_train, y\_test

def generateModel(): #method to read dataset values which contains all five features data

global X, Y, X\_train, X\_test, y\_train, y\_test

train = pd.read\_csv(filename)

X, Y, X\_train, X\_test, y\_train, y\_test = traintest(train)

text.insert(END,"Train & Test Model Generated\n\n")

text.insert(END,"Total Dataset Size : "+str(len(train))+"\n")

text.insert(END,"Split Training Size : "+str(len(X\_train))+"\n")

text.insert(END,"Split Test Size : "+str(len(X\_test))+"\n")

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");

def prediction(X\_test, cls): #prediction done here

y\_pred = cls.predict(X\_test)

for i in range(50):

print("X=%s, Predicted=%s" % (X\_test[i], y\_pred[i]))

return y\_pred

# Function to calculate accuracy

def cal\_accuracy(y\_test, y\_pred, details):

accuracy = accuracy\_score(y\_test,y\_pred)\*100

text.insert(END,details+"\n\n")

text.insert(END,"Accuracy : "+str(accuracy)+"\n\n")

return accuracy

def runRandomForest():

headers = ["Time","V1","V2","V3","V4","V5","V6","V7","V8","V9","V10","V11","V12","V13","V14","V15","V16","V17","V18","V19","V20","V21","V22","V23","V24","V25","V26","V27","V28","Amount","Class"]

global random\_acc

global cls

global X, Y, X\_train, X\_test, y\_train, y\_test

cls = RandomForestClassifier(n\_estimators=50,max\_depth=2,random\_state=0,class\_weight='balanced')

cls.fit(X\_train, y\_train)

text.insert(END,"Prediction Results\n\n")

prediction\_data = prediction(X\_test, cls)

random\_acc = cal\_accuracy(y\_test, prediction\_data,'Random Forest Accuracy')

#str\_tree = export\_graphviz(cls, out\_file=None, feature\_names=headers,filled=True, special\_characters=True, rotate=True, precision=0.6)

#display.display(str\_tree)

def predicts():

global clean

global attack

global total

clean = 0;

attack = 0;

text.delete('1.0', END)

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

test = pd.read\_csv(filename)

test = test.values[:, 0:29]

total = len(test)

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

y\_pred = cls.predict(test)

for i in range(len(test)):

if str(y\_pred[i]) == '1.0':

attack = attack + 1

text.insert(END,"X=%s, Predicted = %s" % (test[i], 'Contains Fraud Transaction Signature')+"\n\n")

else:

clean = clean + 1

text.insert(END,"X=%s, Predicted = %s" % (test[i], 'Transaction Contains Cleaned Signatures')+"\n\n")

def graph():

height = [total,clean,attack]

bars = ('Total Transactions','Normal Transaction','Fraud Transaction')

y\_pos = np.arange(len(bars))

plt.bar(y\_pos, height)

plt.xticks(y\_pos, bars)

plt.show()

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

title = Label(main, text='Fraud Detection Using ML Algorithm')

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 Dataset", command=upload)

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

uploadButton.config(font=font1)

modelButton = Button(main, text="Generate Train & Test Model", command=generateModel)

modelButton.place(x=350,y=550)

modelButton.config(font=font1)

runrandomButton = Button(main, text="Run Random Forest Algorithm", command=runRandomForest)

runrandomButton.place(x=650,y=550)

runrandomButton.config(font=font1)

predictButton = Button(main, text="Detect Fraud From Test Data", command=predicts)

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

predictButton.config(font=font1)

graphButton = Button(main, text="Clean & Fraud Transaction Detection Graph", command=graph)

graphButton.place(x=350,y=600)

graphButton.config(font=font1)

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

exitButton.place(x=770,y=600)

exitButton.config(font=font1)

main.config(bg='LightSkyBlue')

main.mainloop()