

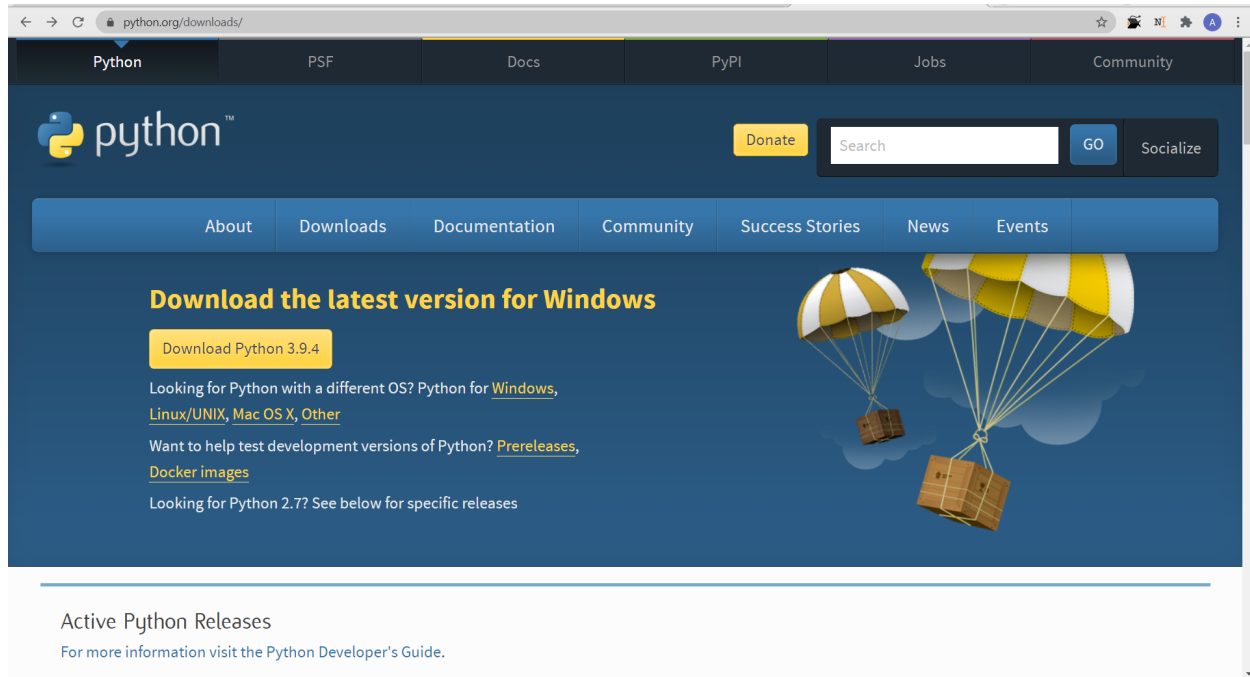
INSTALLATION STEPS

Detection-of-employee-stress-using-ML

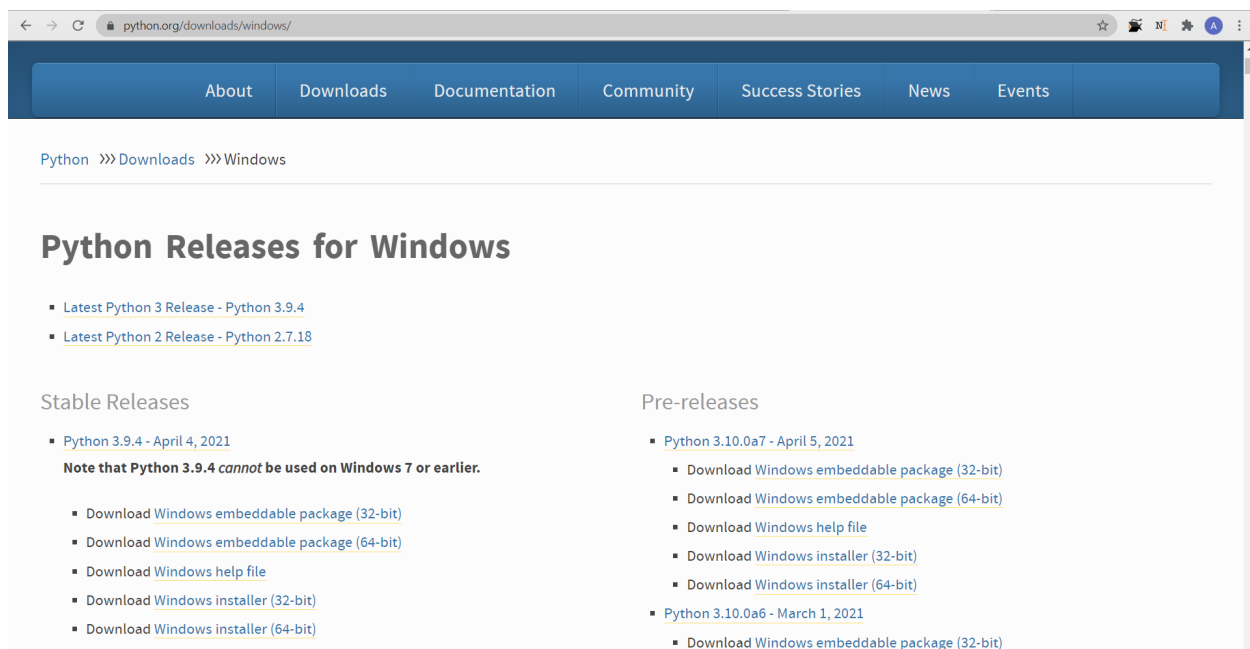
B16-BATCH-06

GITAM DEEMED TO BE UNIVERSITY VIZAG

We are using python3.7.0 version downloaded from the python.org



In python.org we are installing python 3.7.0 version for windows



Note that Python 3.7.0 *cannot* be used on Windows XP or earlier.

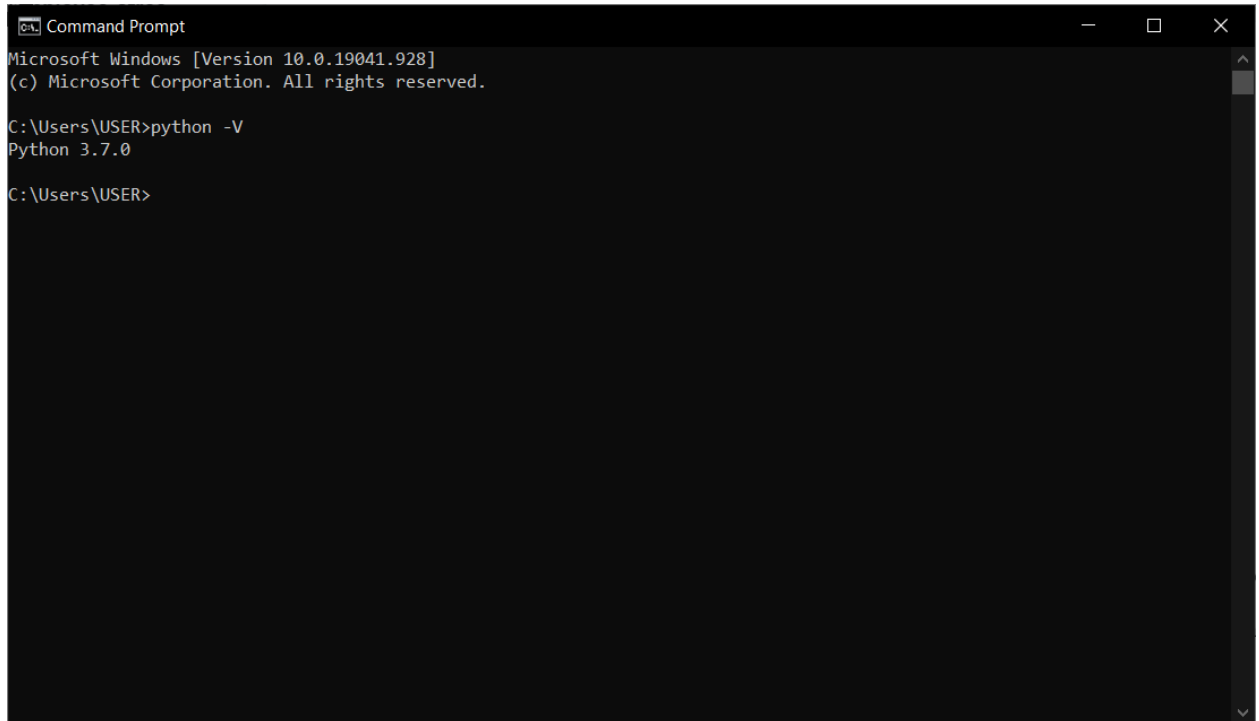
- Download [Windows help file](#)
- Download [Windows x86-64 embeddable zip file](#)
- Download [Windows x86-64 executable installer](#)
- Download [Windows x86-64 web-based installer](#)
- Download [Windows x86 embeddable zip file](#)
- Download [Windows x86 executable installer](#)
- Download [Windows x86 web-based installer](#)
- [Python 3.6.6 - June 27, 2018](#)

We are installing windowsx86 executable installer.

After installing the python and making a path we have to install packages that are required to run the code.

Open the cmd prompt and check if the python is installed or not.

Using the command `python -V`.



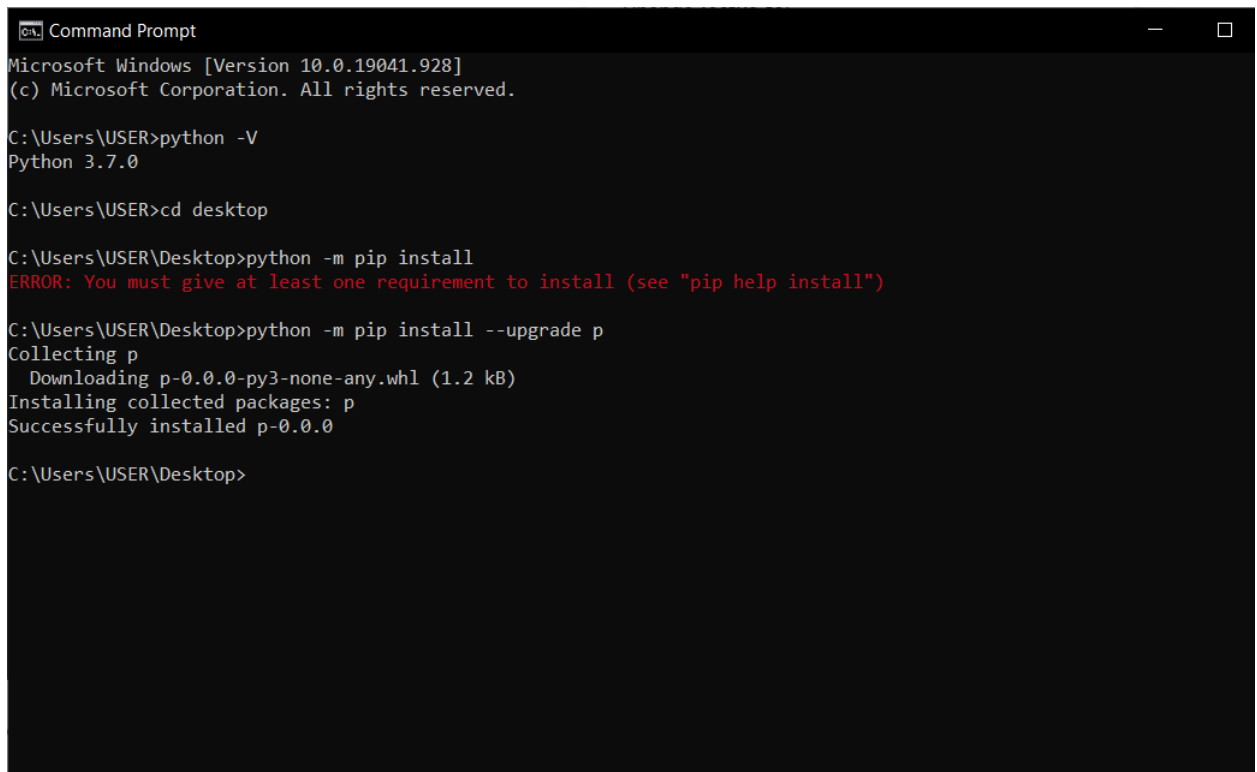
```
Command Prompt
Microsoft Windows [Version 10.0.19041.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\USER>python -V
Python 3.7.0

C:\Users\USER>
```

Change the directory to ,where the code is stored.

Install the pip



```
Command Prompt
Microsoft Windows [Version 10.0.19041.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\USER>python -V
Python 3.7.0

C:\Users\USER>cd desktop

C:\Users\USER\Desktop>python -m pip install
ERROR: You must give at least one requirement to install (see "pip help install")

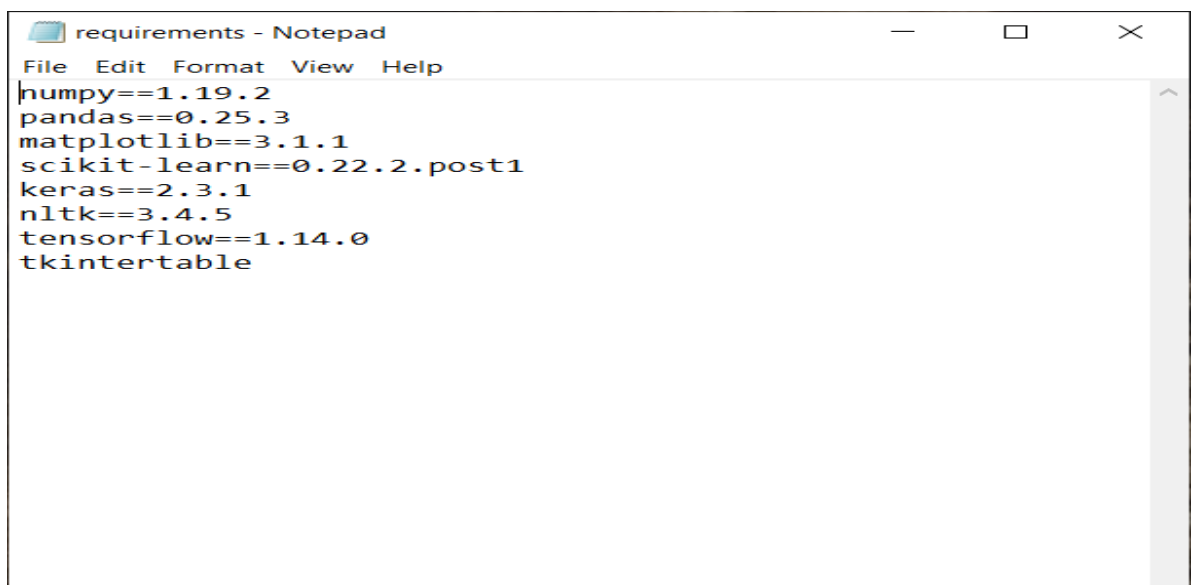
C:\Users\USER\Desktop>python -m pip install --upgrade p
Collecting p
  Downloading p-0.0.0-py3-none-any.whl (1.2 kB)
Installing collected packages: p
Successfully installed p-0.0.0

C:\Users\USER\Desktop>
```

NOW we have to install all the packages that we need

Open the notepad and write the packages

And save file



```
requirements - Notepad
File Edit Format View Help
numpy==1.19.2
pandas==0.25.3
matplotlib==3.1.1
scikit-learn==0.22.2.post1
keras==2.3.1
nltk==3.4.5
tensorflow==1.14.0
tkintertable
```

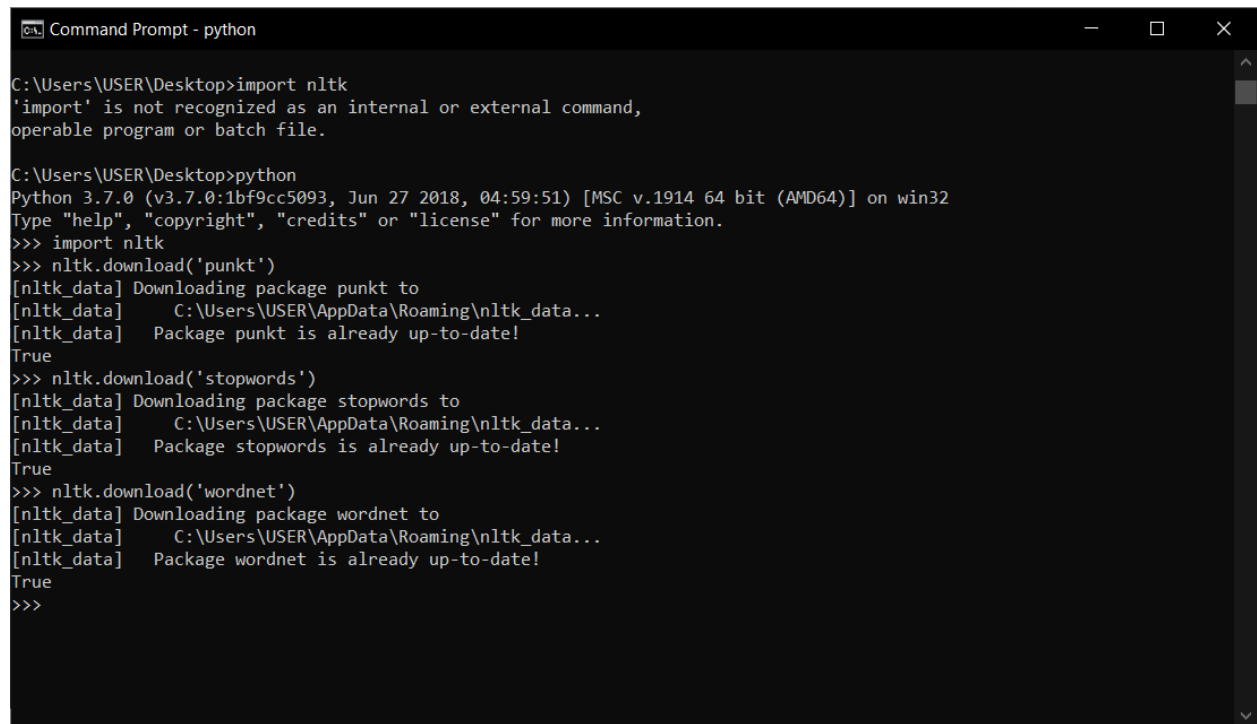
Now write the command

`pip install -r requirements.txt`

```
C:\Users\USER\Desktop>pip install -r requirements.txt
Requirement already satisfied: numpy==1.19.2 in c:\python37\lib\site-packages (from -r requirements.txt (line 1)) (1.19.2)
Requirement already satisfied: pandas==0.25.3 in c:\python37\lib\site-packages (from -r requirements.txt (line 2)) (0.25.3)
Requirement already satisfied: matplotlib==3.1.1 in c:\python37\lib\site-packages (from -r requirements.txt (line 3)) (3.1.1)
Requirement already satisfied: scikit-learn==0.22.2.post1 in c:\python37\lib\site-packages (from -r requirements.txt (line 4)) (0.22.2.post1)
Requirement already satisfied: keras==2.3.1 in c:\python37\lib\site-packages (from -r requirements.txt (line 5)) (2.3.1)
Requirement already satisfied: nltk==3.4.5 in c:\python37\lib\site-packages (from -r requirements.txt (line 6)) (3.4.5)
Requirement already satisfied: tensorflow==1.14.0 in c:\python37\lib\site-packages (from -r requirements.txt (line 7)) (1.14.0)
Requirement already satisfied: tkinter in c:\python37\lib\site-packages (from -r requirements.txt (line 8)) (1.3.2)
Requirement already satisfied: pytz==2017.2 in c:\python37\lib\site-packages (from pandas==0.25.3->-r requirements.txt (line 2)) (2017.2)
```

```
Command Prompt
Requirement already satisfied: h5py in c:\python37\lib\site-packages (from keras==2.3.1->-r requirements.txt (line 5)) (3.2.1)
Requirement already satisfied: keras-preprocessing>=1.0.5 in c:\python37\lib\site-packages (from keras==2.3.1->-r requirements.txt (line 5)) (1.1.2)
Requirement already satisfied: pyyaml in c:\python37\lib\site-packages (from keras==2.3.1->-r requirements.txt (line 5)) (5.4.1)
Requirement already satisfied: keras-applications>=1.0.6 in c:\python37\lib\site-packages (from keras==2.3.1->-r requirements.txt (line 5)) (1.0.8)
Requirement already satisfied: protobuf>=3.6.1 in c:\python37\lib\site-packages (from tensorflow==1.14.0->-r requirements.txt (line 7)) (3.15.8)
Requirement already satisfied: tensorflow-estimator<1.15.0rc0,>=1.14.0rc0 in c:\python37\lib\site-packages (from tensorflow==1.14.0->-r requirements.txt (line 7)) (1.14.0)
Requirement already satisfied: gast>=0.2.0 in c:\python37\lib\site-packages (from tensorflow==1.14.0->-r requirements.txt (line 7)) (0.4.0)
Requirement already satisfied: astor>=0.6.0 in c:\python37\lib\site-packages (from tensorflow==1.14.0->-r requirements.txt (line 7)) (0.8.1)
Requirement already satisfied: termcolor>=1.1.0 in c:\python37\lib\site-packages (from tensorflow==1.14.0->-r requirements.txt (line 7)) (1.1.0)
Requirement already satisfied: tensorboard<1.15.0,>=1.14.0 in c:\python37\lib\site-packages (from tensorflow==1.14.0->-r requirements.txt (line 7)) (1.14.0)
Requirement already satisfied: grpcio>=1.8.6 in c:\python37\lib\site-packages (from tensorflow==1.14.0->-r requirements.txt (line 7)) (1.37.1)
Requirement already satisfied: wheel>=0.26 in c:\python37\lib\site-packages (from tensorflow==1.14.0->-r requirements.txt (line 7)) (0.36.2)
Requirement already satisfied: google-pasta>=0.1.6 in c:\python37\lib\site-packages (from tensorflow==1.14.0->-r requirements.txt (line 7)) (0.2.0)
Requirement already satisfied: wrapt>=1.11.1 in c:\python37\lib\site-packages (from tensorflow==1.14.0->-r requirements.txt (line 7)) (1.12.1)
Requirement already satisfied: absl-py>=0.7.0 in c:\python37\lib\site-packages (from tensorflow==1.14.0->-r requirements.txt (line 7)) (0.12.0)
```

I had already installed it so it is showing that requirement already satisfied.



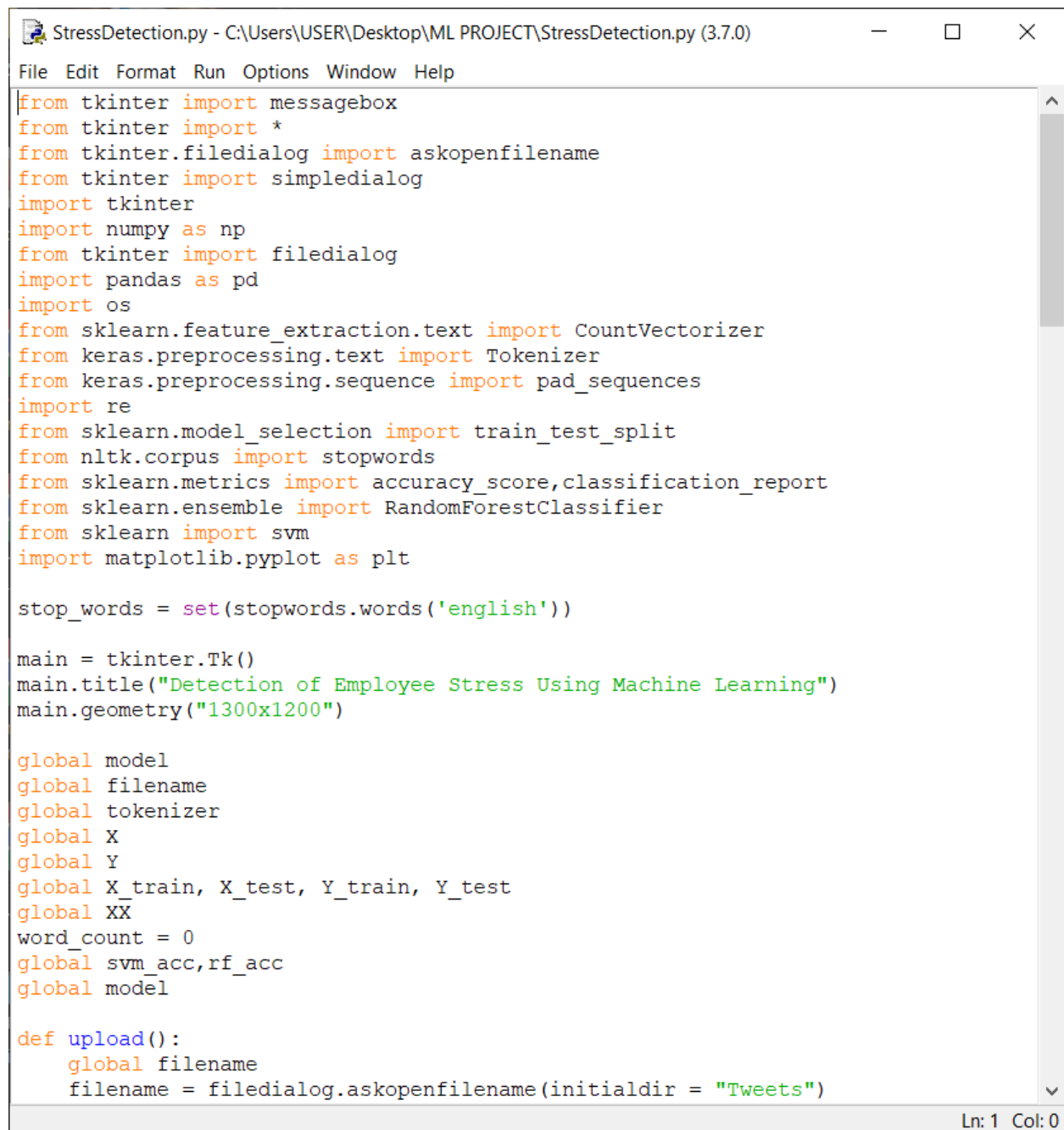
```
Command Prompt - python

C:\Users\USER\Desktop>import nltk
'import' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\USER\Desktop>python
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import nltk
>>> nltk.download('punkt')
[nltk_data] Downloading package punkt to
[nltk_data]   C:\Users\USER\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
True
>>> nltk.download('stopwords')
[nltk_data] Downloading package stopwords to
[nltk_data]   C:\Users\USER\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
True
>>> nltk.download('wordnet')
[nltk_data] Downloading package wordnet to
[nltk_data]   C:\Users\USER\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
True
>>>
```

All the requirements for the code are satisfied.

Open the code and run the code



```
StressDetection.py - C:\Users\USER\Desktop\ML PROJECT\StressDetection.py (3.7.0)
File Edit Format Run Options Window Help
from tkinter import messagebox
from tkinter import *
from tkinter.filedialog import askopenfilename
from tkinter import simpledialog
import tkinter
import numpy as np
from tkinter import filedialog
import pandas as pd
import os
from sklearn.feature_extraction.text import CountVectorizer
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
import re
from sklearn.model_selection import train_test_split
from nltk.corpus import stopwords
from sklearn.metrics import accuracy_score, classification_report
from sklearn.ensemble import RandomForestClassifier
from sklearn import svm
import matplotlib.pyplot as plt

stop_words = set(stopwords.words('english'))

main = tkinter.Tk()
main.title("Detection of Employee Stress Using Machine Learning")
main.geometry("1300x1200")

global model
global filename
global tokenizer
global X
global Y
global X_train, X_test, Y_train, Y_test
global XX
word_count = 0
global svm_acc, rf_acc
global model

def upload():
    global filename
    filename = filedialog.askopenfilename(initialdir = "Tweets")
```

Ln: 1 Col: 0


```

StressDetection.py - C:\Users\USER\Desktop\ML PROJECT\StressDetection.py (3.7.0)
File Edit Format Run Options Window Help
global model

def upload():
    global filename
    filename = filedialog.askopenfilename(initialdir = "Tweets")
    pathlabel.config(text=filename)
    textarea.delete('1.0', END)
    textarea.insert(END, 'tweets dataset loaded\n')

def preprocess():
    global X
    global Y
    global word_count
    X = []
    Y = []
    textarea.delete('1.0', END)
    train = pd.read_csv(filename, encoding='iso-8859-1')
    word_count = 0
    words = []
    for i in range(len(train)):
        label = train.get_value(i, 2, takeable = True)
        tweet = train.get_value(i, 1, takeable = True)
        tweet = tweet.lower()
        arr = tweet.split(" ")
        msg = ''
        for k in range(len(arr)):
            word = arr[k].strip()
            if len(word) > 2 and word not in stop_words:
                msg += word + " "
                if word not in words:
                    words.append(word)
        text = msg.strip()
        X.append(text)
        Y.append(int(label))
    X = np.asarray(X)
    Y = np.asarray(Y)
    word_count = len(words)
    textarea.insert(END, 'Total tweets found in dataset : '+str(len(X))+"\n")
    textarea.insert(END, 'Total words found in all tweets : '+str(len(words))+"\n")

```

Ln: 40 Col: 0

```

StressDetection.py - C:\Users\USER\Desktop\ML PROJECT\StressDetection.py (3.7.0)
File Edit Format Run Options Window Help

def featureExtraction():
    global X
    global Y
    global XX
    global tokenizer
    global X_train, X_test, Y_train, Y_test
    max_fatures = word_count
    tokenizer = Tokenizer(num_words=max_fatures, split=' ')
    tokenizer.fit_on_texts(X)
    XX = tokenizer.texts_to_sequences(X)
    XX = pad_sequences(XX)
    indices = np.arange(XX.shape[0])
    np.random.shuffle(indices)
    XX = XX[indices]
    Y = Y[indices]
    X_train, X_test, Y_train, Y_test = train_test_split(XX, Y, test_size = 0.13,
    textarea.insert(END, 'Total features extracted from tweets are : '+str(X_train.shape[0]))
    textarea.insert(END, 'Total splitted records used for training : '+str(len(X_train)))
    textarea.insert(END, 'Total splitted records used for testing : '+str(len(X_test)))

def SVM():
    textarea.delete('1.0', END)
    global svm_acc
    rfc = svm.SVC(C=2.0, gamma='scale', kernel = 'rbf', random_state = 2)
    rfc.fit(X_train, Y_train)
    textarea.insert(END, "SVM Prediction Results\n")
    prediction_data = rfc.predict(X_test)
    svm_acc = accuracy_score(Y_test, prediction_data)*100
    classf_svm = str(classification_report(Y_test, prediction_data))
    textarea.insert(END, "SVM Accuracy : "+str(svm_acc)+"\n\n" + classf_svm)

def RandomForest():
    global rf_acc
    global model
    rfc = RandomForestClassifier(n_estimators=20, random_state=0)
    rfc.fit(X_train, Y_train)
    textarea.insert(END, "Random Forest Prediction Results\n")
    prediction_data = rfc.predict(X_test)
    rf_acc = accuracy_score(Y_test, prediction_data)*100
    classf = str(classification_report(Y_test, prediction_data))

```

Ln: 75 Col: 79

```

StressDetection.py - C:\Users\USER\Desktop\ML PROJECT\StressDetection.py (3.7.0)
File Edit Format Run Options Window Help

def predict():
    textarea.delete('1.0', END)
    testfile = filedialog.askopenfilename(initialdir = "Tweets")
    test = pd.read_csv(testfile,encoding='iso-8859-1')

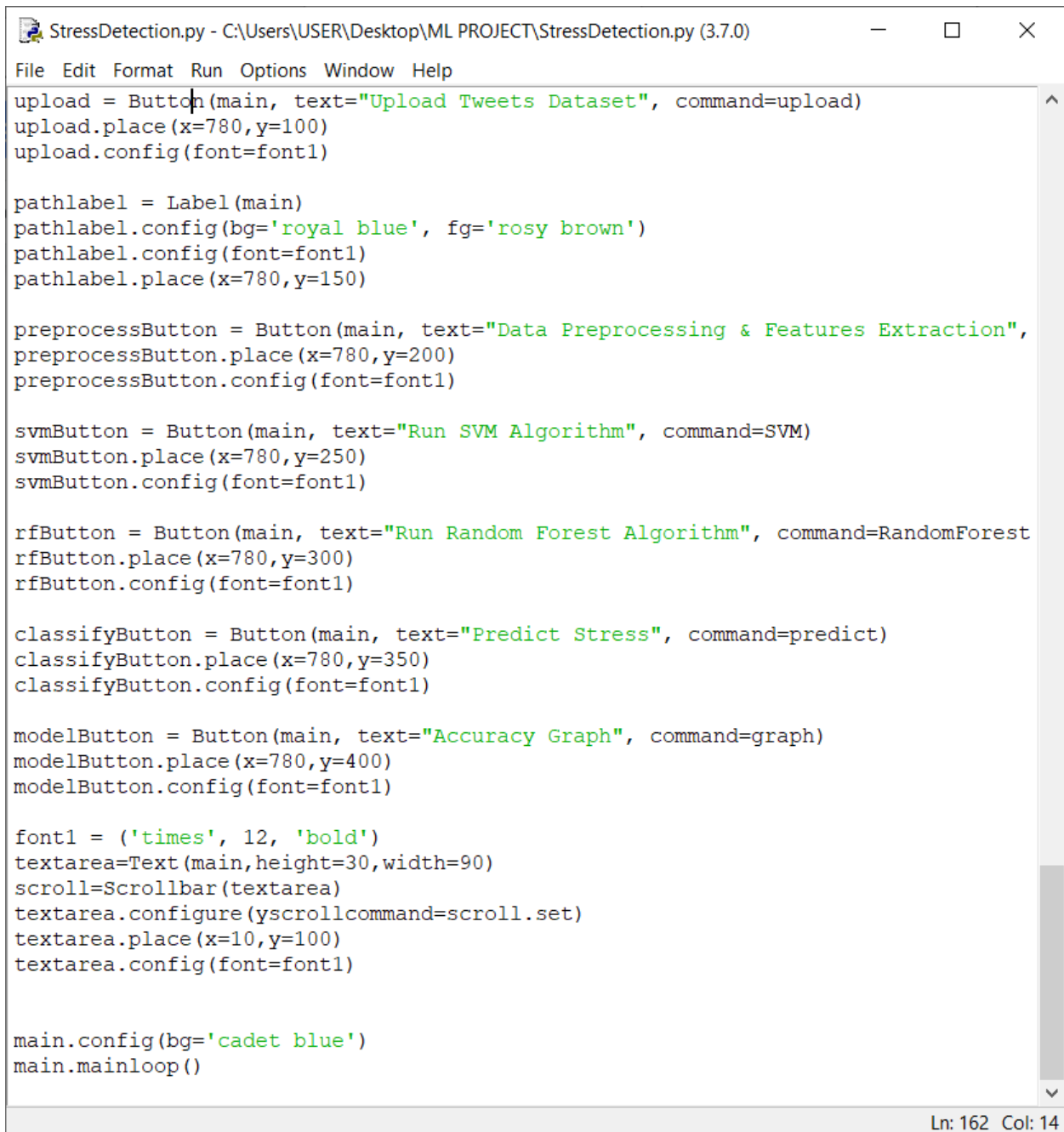
    for i in range(len(test)):
        tweet = test.get_value(i,0,takeable = True)
        arr = tweet.split(" ")
        msg = ''
        for j in range(len(arr)):
            word = arr[j].strip()
            if len(word) > 2 and word not in stop_words:
                msg+=word+" "
        text = msg.strip()
        mytext = [text]
        twts = tokenizer.texts_to_sequences(mytext)
        twts = pad_sequences(twts, maxlen=83, dtype='int32', value=0)
        stress = model.predict(twts)
        print(stress)
        if stress == 0:
            textarea.insert(END,text+' == Prediction Result : Not Stressed\n\n')
        if stress == 1:
            textarea.insert(END,text+' == Prediction Result : Stressed\n\n')

def graph():
    height = [svm_acc,rf_acc]
    bars = ('SVM ACC','Random Forest ACC')
    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='Detection of Employee Stress Using Machine Learning')
title.config(bg='yellow green', fg='saddle brown')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)

```

Ln: 117 Col: 62



```

StressDetection.py - C:\Users\USER\Desktop\ML PROJECT\StressDetection.py (3.7.0)
File Edit Format Run Options Window Help
upload = Button(main, text="Upload Tweets Dataset", command=upload)
upload.place(x=780,y=100)
upload.config(font=font1)

pathlabel = Label(main)
pathlabel.config(bg='royal blue', fg='rosy brown')
pathlabel.config(font=font1)
pathlabel.place(x=780,y=150)

preprocessButton = Button(main, text="Data Preprocessing & Features Extraction",
preprocessButton.place(x=780,y=200)
preprocessButton.config(font=font1)

svmButton = Button(main, text="Run SVM Algorithm", command=SVM)
svmButton.place(x=780,y=250)
svmButton.config(font=font1)

rfButton = Button(main, text="Run Random Forest Algorithm", command=RandomForest
rfButton.place(x=780,y=300)
rfButton.config(font=font1)

classifyButton = Button(main, text="Predict Stress", command=predict)
classifyButton.place(x=780,y=350)
classifyButton.config(font=font1)

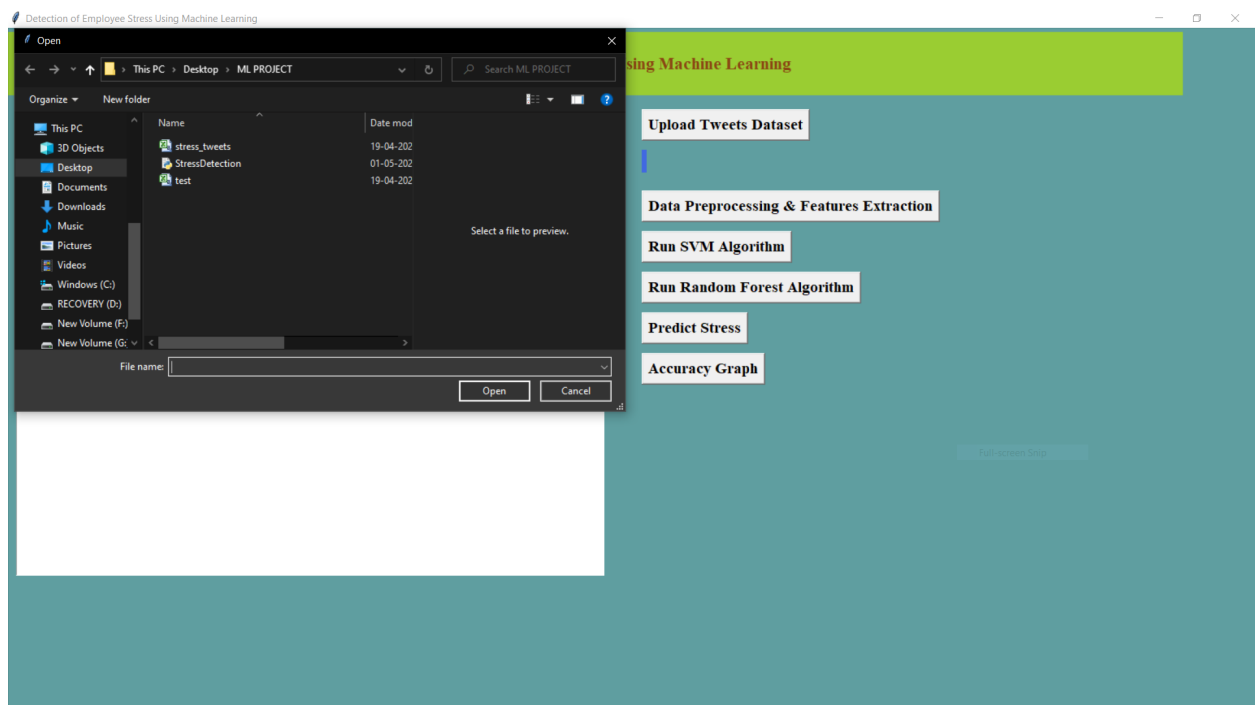
modelButton = Button(main, text="Accuracy Graph", command=graph)
modelButton.place(x=780,y=400)
modelButton.config(font=font1)

font1 = ('times', 12, 'bold')
textarea=Text(main,height=30,width=90)
scroll=Scrollbar(textarea)
textarea.configure(yscrollcommand=scroll.set)
textarea.place(x=10,y=100)
textarea.config(font=font1)

main.config(bg='cadet blue')
main.mainloop()
Ln: 162 Col: 14

```

Output :



Upload the data to train and test that we initially collected.



Click the Data Preprocessing & Feature Extraction

Detection of Employee Stress Using Machine Learning

Total tweets found in dataset : 10314
Total words found in all tweets : 30790

Total features extracted from tweets are : 83
Total splitted records used for training : 8973
Total splitted records used for testing : 1341

Upload Tweets Dataset

C:/Users/USER/Desktop/ML PROJECT/stress_tweets.csv

Data Preprocessing & Features Extraction

Run SVM Algorithm

Run Random Forest Algorithm

Predict Stress

Accuracy Graph

Result of SVM algorithm and Run Random Forest Algorithm

Detection of Employee Stress Using Machine Learning

SVM Prediction Results
SVM Accuracy : 91.72259507829978

	precision	recall	f1-score	support
0	0.91	0.99	0.95	1070
1	0.95	0.62	0.75	271

accuracy 0.92 1341
macro avg 0.93 0.81 0.85 1341
weighted avg 0.92 0.92 0.91 1341

Random Forest Prediction Results
Random Forest Accuracy : 99.4034302759135

	precision	recall	f1-score	support
0	0.99	1.00	1.00	1070
1	0.99	0.98	0.99	271

accuracy 0.99 1341
macro avg 0.99 0.99 0.99 1341
weighted avg 0.99 0.99 0.99 1341

Upload Tweets Dataset

C:/Users/USER/Desktop/ML PROJECT/stress_tweets.csv

Data Preprocessing & Features Extraction

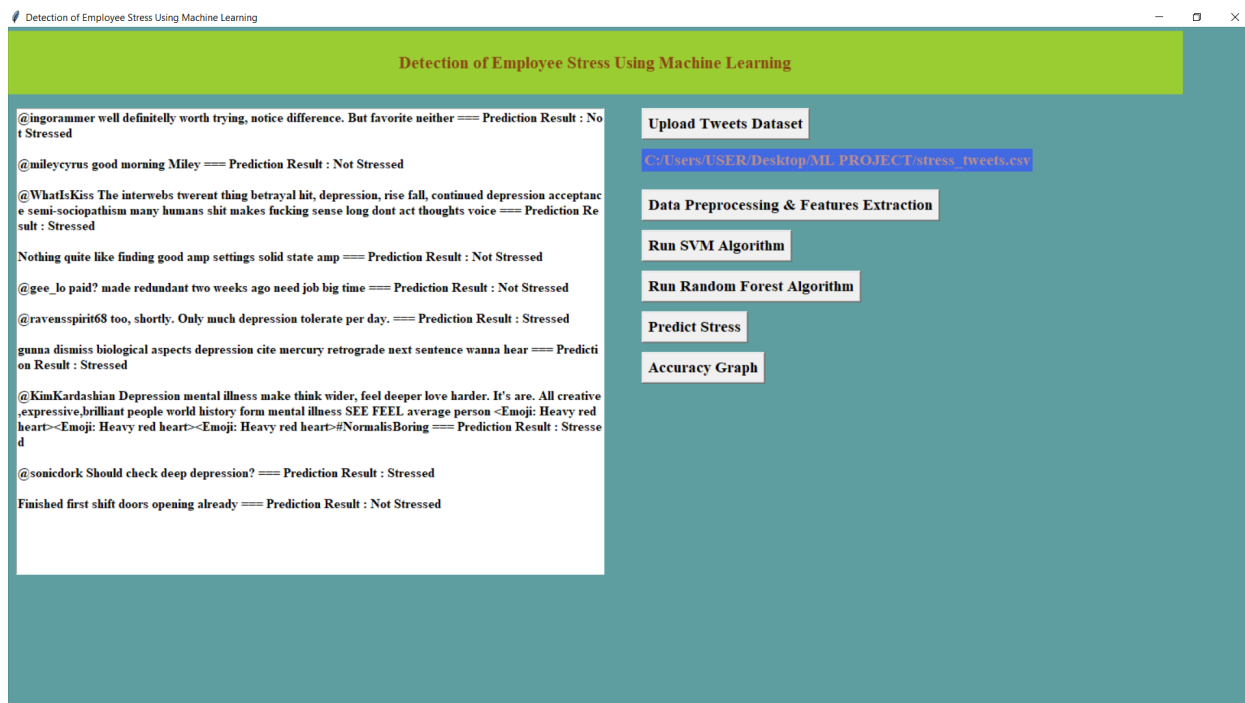
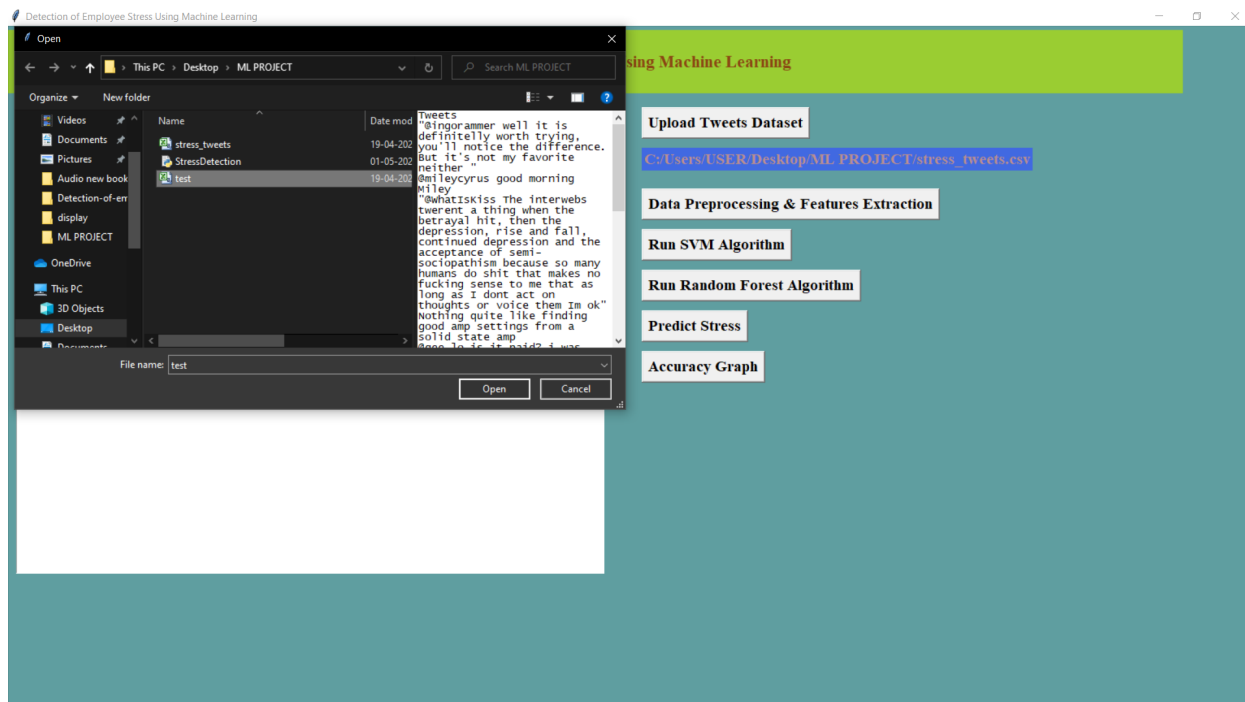
Run SVM Algorithm

Run Random Forest Algorithm

Predict Stress

Accuracy Graph

Find the stress



Accuracy graphs

