****

Session 2021-2022

**Submitted To**:

**Submitted By:** Prof. Punit Kumar Jahary

**Name** - Shivansh Shrivastava

**Enroll** – 0901IT181056

**Subject-** Critical Thinking

**Problem 1:** Now a days we are trying to develop the customer oriented and business oriented products to ease the product utilization. Design an application (GUI) to launch the few frequent applications that users wants.

**Input format:** through text or voice input (Like Alexa).

**Output:** launch the application (open chrome, notepad, media player etc.)

**Language:** Python

**SOLUTION:**

**Python Code:**

import pyttsx3  
import os  
import speech\_recognition  
  
engine = pyttsx3.init()  
voices = engine.getProperty('voices')  
  
engine.setProperty('voice', voices[1].id)  
engine.runAndWait()  
print("")  
print("")  
  
print(" =============================================== Hello World!! ================================================")  
engine.say('Hello World!!')  
  
print("")  
print(" My name is Devansh Sharma,I make this tool With this help of tool you can open below things.......")  
  
print("\n\t 1.MICROSOFT WORD \t 2.MICROSOFT POWERPOINT \n\t 3.MICROSOFT EXCEL \t 4.GOOGLE CHROME \n\t 5.VLC PLAYER \t 6.ADOBE ILLUSTRATOR \n\t 7.ADOBE PHOTOSHOP \t 8.MICROSOFT EDGE \n\t 9.NOTEPAD \t 10.TELEGRAM \n\n\t\t 0. FOR EXIT")  
  
print("\n (YOU CAN USE NUMBER OR YOU CAN DO CHAT LIKE 'OPEN NOTEBOOK' etc....)")  
  
print("\n ============================================ Welcome To My Tools ============================================")  
pyttsx3.speak("Welcome to my tools")  
print("")  
print("")  
  
pyttsx3.speak("chat with me with your requirements")  
  
while True:  
 print(" CHAT WITH ME WITH YOUR REQUIREMENTS : ", end='')  
 p = input()  
 p = p.upper()  
 print(p)  
  
 if ("DONT" in p) or ("DON'T" in p) or ("NOT" in p):  
 pyttsx3.speak("Type Again")  
 print(".")  
 print(".")  
 continue  
  
 elif ("GOOGLE" in p) or ("SEARCH" in p) or ("WEB BROWSER" in p) or ("CHROME" in p) or ("BROWSER" in p) or ("4" in p):  
 pyttsx3.speak("Opening")  
 pyttsx3.speak("GOOGLE CHROME")  
 print(".")  
 print(".")  
 os.system("chrome")  
  
 elif ("IE" in p) or ("MSEDGE" in p) or ("EDGE" in p) or ("8" in p):  
 pyttsx3.speak("Opening")  
 pyttsx3.speak("MICROSOFT EDGE")  
 print(".")  
 print(".")  
 os.system("msedge")  
  
 elif ("NOTE" in p) or ("NOTES" in p) or ("NOTEPAD" in p) or ("EDITOR" in p) or ("9" in p):  
 pyttsx3.speak("Opening")  
 pyttsx3.speak("NOTEPAD")  
 print(".")  
 print(".")  
 os.system("Notepad")  
  
 elif ("VLCPLAYER" in p) or ("PLAYER" in p) or ("VIDEO PLAYER" in p) or ("5" in p):  
 pyttsx3.speak("Opening")  
 pyttsx3.speak("VLC PLAYER")  
 print(".")  
 print(".")  
 os.system("VLC")  
  
 elif ("ILLUSTRATOR" in p) or ("AI" in p) or ("6" in p):  
 pyttsx3.speak("Opening")  
 pyttsx3.speak("ADOBE ILLUSTRATOR")  
 print(".")  
 print(".")  
 os.system("illustrator")  
  
 elif ("PHOTOSHOP" in p) or ("PS" in p) or ("PHOTOSHOP CC" in p) or ("7" in p):  
 pyttsx3.speak("Opening")  
 pyttsx3.speak("ADOBE PHOTOSHOP")  
 print(".")  
 print(".")  
 os.system("photoshop")  
  
 elif ("TELEGRAM" in p) or ("TG" in p) or ("10" in p):  
 pyttsx3.speak("Opening")  
 pyttsx3.speak("TELEGRAM")  
 print(".")  
 print(".")  
 os.system("telegram")  
  
 elif ("EXCEL" in p) or ("MSEXCEL" in p) or ("SHEET" in p) or ("WINEXCEL" in p) or ("3" in p):  
 pyttsx3.speak("Opening")  
 pyttsx3.speak("MICROSOFT EXCEL")  
 print(".")  
 print(".")  
 os.system("excel")  
  
 elif ("SLIDE" in p) or ("MSPOWERPOINT" in p) or ("PPT" in p) or ("POWERPNT" in p) or ("2" in p):  
 pyttsx3.speak("Opening")  
 pyttsx3.speak("MICROSOFT POWERPOINT")  
 print(".")  
 print(".")  
 os.system("powerpnt")  
  
 elif ("WORD" in p) or ("MSWORD" in p) or ("1" in p):  
 pyttsx3.speak("Opening")  
 pyttsx3.speak("MICROSOFT WORD")  
 print(".")  
 print(".")  
 os.system("winword")  
  
  
 elif ("EXIT" in p) or ("QUIT" in p) or ("CLOSE" in p) or ("0" in p):  
 pyttsx3.speak("Exiting")  
 break  
  
 else:  
 pyttsx3.speak(p)  
 print("Is Invalid,Please Try Again")  
 pyttsx3.speak("is Invalid,Please try again")  
 print(".")  
 print(".")

**Output:**

**Text

Description automatically generated**

**Problem 2:**

MNIST ("Modified National Institute of Standards and Technology") is the de facto “hello world” dataset of computer vision. This classic dataset of handwritten images has served as the basis for benchmarking classification algorithms. As new machine learning techniques emerge, MNIST remains a reliable resource for researchers and learners alike.

Your goal is to correctly identify digits from a dataset of tens of thousands of handwritten images. We encourage you to experiment with different algorithms to learn first-hand what works well and how techniques compare.

**SOLUTION:**

# **Load MNIST Data**

**IN [0]:**

# MNIST dataset downloaded from Kaggle :

#https://www.kaggle.com/c/digit-recognizer/data

# Functions to read and show images.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

d0 = pd.read\_csv('./mnist\_train.csv')

print(d0.head(5)) # print first five rows of d0.

# save the labels into a variable l.

l = d0['label']

# Drop the label feature and store the pixel data in d.

d = d0.drop("label",axis=1)

label pixel0 pixel1 pixel2 pixel3 pixel4 pixel5 pixel6 pixel7 \

0 1 0 0 0 0 0 0 0 0

1 0 0 0 0 0 0 0 0 0

2 1 0 0 0 0 0 0 0 0

3 4 0 0 0 0 0 0 0 0

4 0 0 0 0 0 0 0 0 0

pixel8 ... pixel774 pixel775 pixel776 pixel777 pixel778 \

0 0 ... 0 0 0 0 0

1 0 ... 0 0 0 0 0

2 0 ... 0 0 0 0 0

3 0 ... 0 0 0 0 0

4 0 ... 0 0 0 0 0

pixel779 pixel780 pixel781 pixel782 pixel783

0 0 0 0 0 0

1 0 0 0 0 0

2 0 0 0 0 0

3 0 0 0 0 0

4 0 0 0 0 0

[5 rows x 785 columns]

In [0]:

print(d**.**shape)

print(l**.**shape)

(42000, 784)

(42000,)

In [0]:

*# display or plot a number.*

plt**.**figure(figsize**=**(7,7))

idx **=** 1

grid\_data **=** d**.**iloc[idx]**.**as\_matrix()**.**reshape(28,28) *# reshape from 1d to 2d pixel array*

plt**.**imshow(grid\_data, interpolation **=** "none", cmap **=** "gray")

plt**.**show()

print(l[idx])

A picture containing text, clipart

Description automatically generated

# **2D Visualization using PCA**

**IN [0]:**

*# Pick first 15K data-points to work on for time-effeciency.*

*#Excercise: Perform the same analysis on all of 42K data-points.*

labels **=** l**.**head(15000)

data **=** d**.**head(15000)

print("the shape of sample data = ", data**.**shape)

the shape of sample data = (15000, 784)

**IN [0]:**

*# Data-preprocessing: Standardizing the data*

**from** sklearn.preprocessing **import** StandardScaler

standardized\_data **=** StandardScaler()**.**fit\_transform(data)

print(standardized\_data**.**shape)

(15000, 784)

**IN [0]:**

*#find the co-variance matrix which is : A^T \* A*

sample\_data **=** standardized\_data

*# matrix multiplication using numpy*

covar\_matrix **=** np**.**matmul(sample\_data**.**T , sample\_data)

print ( "The shape of variance matrix = ", covar\_matrix**.**shape)

The shape of variance matrix = (784, 784)

**IN [0]:**

*# finding the top two eigen-values and corresponding eigen-vectors*

*# for projecting onto a 2-Dim space.*

**from** scipy.linalg **import** eigh

*# the parameter 'eigvals' is defined (low value to heigh value)*

*# eigh function will return the eigen values in asending order*

*# this code generates only the top 2 (782 and 783) eigenvalues.*

values, vectors **=** eigh(covar\_matrix, eigvals**=**(782,783))

print("Shape of eigen vectors = ",vectors**.**shape)

*# converting the eigen vectors into (2,d) shape for easyness of further computations*

vectors **=** vectors**.**T

print("Updated shape of eigen vectors = ",vectors**.**shape)

*# here the vectors[1] represent the eigen vector corresponding 1st principal eigen vector*

*# here the vectors[0] represent the eigen vector corresponding 2nd principal eigen vector*

Shape of eigen vectors = (784, 2)

Updated shape of eigen vectors = (2, 784)

**IN [0]:**

*# projecting the original data sample on the plane*

*#formed by two principal eigen vectors by vector-vector multiplication.*

**import** matplotlib.pyplot **as** plt

new\_coordinates **=** np**.**matmul(vectors, sample\_data**.**T)

print (" resultanat new data points' shape ", vectors**.**shape, "X", sample\_data**.**T**.**shape," = ", new\_coordinates**.**shape)

resultanat new data points' shape (2, 784) X (784, 15000) = (2, 15000)

**IN [0]:**

**import** pandas **as** pd

*# appending label to the 2d projected data*

new\_coordinates **=** np**.**vstack((new\_coordinates, labels))**.**T

*# creating a new data frame for ploting the labeled points.*

dataframe **=** pd**.**DataFrame(data**=**new\_coordinates, columns**=**("1st\_principal", "2nd\_principal", "label"))

print(dataframe**.**head())

1st\_principal 2nd\_principal label

0 -5.558661 -5.043558 1.0

1 6.193635 19.305278 0.0

2 -1.909878 -7.678775 1.0

3 5.525748 -0.464845 4.0

4 6.366527 26.644289 0.0

In [0]:

**IN [0]:**

*# ploting the 2d data points with seaborn*

**import** seaborn **as** sn

sn**.**FacetGrid(dataframe, hue**=**"label", size**=**6)**.**map(plt**.**scatter, '1st\_principal', '2nd\_principal')**.**add\_legend()

plt**.**show()

Chart, scatter chart

Description automatically generated

**PCA USING Scikit-Learn**

**IN [0]**

*# initializing the pca*

**from** sklearn **import** decomposition

pca **=** decomposition**.**PCA()

**IN [0]**

*# configuring the parameteres*

*# the number of components = 2*

pca**.**n\_components **=** 2

pca\_data **=** pca**.**fit\_transform(sample\_data)

*# pca\_reduced will contain the 2-d projects of simple data*

print("shape of pca\_reduced.shape = ", pca\_data**.**shape)

shape of pca\_reduced.shape = (15000, 2)

**IN [0]**

*# attaching the label for each 2-d data point*

pca\_data **=** np**.**vstack((pca\_data**.**T, labels))**.**T

*# creating a new data fram which help us in ploting the result data*

pca\_df **=** pd**.**DataFrame(data**=**pca\_data, columns**=**("1st\_principal", "2nd\_principal", "label"))

sn**.**FacetGrid(pca\_df, hue**=**"label", size**=**6)**.**map(plt**.**scatter, '1st\_principal', '2nd\_principal')**.**add\_legend()

plt**.**show()

**Chart, scatter chart

Description automatically generated**

# **PCA for dimensionality redcution (not for visualization)**

**IN [0]**

*# PCA for dimensionality redcution (non-visualization)*

pca**.**n\_components **=** 784

pca\_data **=** pca**.**fit\_transform(sample\_data)

percentage\_var\_explained **=** pca**.**explained\_variance\_ **/** np**.**sum(pca**.**explained\_variance\_);

cum\_var\_explained **=** np**.**cumsum(percentage\_var\_explained)

*# Plot the PCA spectrum*

plt**.**figure(1, figsize**=**(6, 4))

plt**.**clf()

plt**.**plot(cum\_var\_explained, linewidth**=**2)

plt**.**axis('tight')

plt**.**grid()

plt**.**xlabel('n\_components')

plt**.**ylabel('Cumulative\_explained\_variance')

plt**.**show()

*# If we take 200-dimensions, approx. 90% of variance is expalined.*

**A picture containing line chart

Description automatically generated**

**t-SNE using Scikit-Learn**

**IN [0]**

*# TSNE*

**from** sklearn.manifold **import** TSNE

*# Picking the top 1000 points as TSNE takes a lot of time for 15K points*

data\_1000 **=** standardized\_data[0:1000,:]

labels\_1000 **=** labels[0:1000]

model **=** TSNE(n\_components**=**2, random\_state**=**0)

*# configuring the parameteres*

*# the number of components = 2*

*# default perplexity = 30*

*# default learning rate = 200*

*# default Maximum number of iterations for the optimization = 1000*

tsne\_data **=** model**.**fit\_transform(data\_1000)

*# creating a new data frame which help us in ploting the result data*

tsne\_data **=** np**.**vstack((tsne\_data**.**T, labels\_1000))**.**T

tsne\_df **=** pd**.**DataFrame(data**=**tsne\_data, columns**=**("Dim\_1", "Dim\_2", "label"))

*# Ploting the result of tsne*

sn**.**FacetGrid(tsne\_df, hue**=**"label", size**=**6)**.**map(plt**.**scatter, 'Dim\_1', 'Dim\_2')**.**add\_legend()

plt**.**show()

Chart, scatter chart

Description automatically generated

**IN [0]**

model **=** TSNE(n\_components**=**2, random\_state**=**0, perplexity**=**50)

tsne\_data **=** model**.**fit\_transform(data\_1000)

*# creating a new data fram which help us in ploting the result data*

tsne\_data **=** np**.**vstack((tsne\_data**.**T, labels\_1000))**.**T

tsne\_df **=** pd**.**DataFrame(data**=**tsne\_data, columns**=**("Dim\_1", "Dim\_2", "label"))

*# Ploting the result of tsne*

sn**.**FacetGrid(tsne\_df, hue**=**"label", size**=**6)**.**map(plt**.**scatter, 'Dim\_1', 'Dim\_2')**.**add\_legend()

plt**.**title('With perplexity = 50')

plt**.**show()

Chart, scatter chart

Description automatically generated

**IN [0]**

model **=** TSNE(n\_components**=**2, random\_state**=**0, perplexity**=**50, n\_iter**=**5000)

tsne\_data **=** model**.**fit\_transform(data\_1000)

*# creating a new data fram which help us in ploting the result data*

tsne\_data **=** np**.**vstack((tsne\_data**.**T, labels\_1000))**.**T

tsne\_df **=** pd**.**DataFrame(data**=**tsne\_data, columns**=**("Dim\_1", "Dim\_2", "label"))

*# Ploting the result of tsne*

sn**.**FacetGrid(tsne\_df, hue**=**"label", size**=**6)**.**map(plt**.**scatter, 'Dim\_1', 'Dim\_2')**.**add\_legend()

plt**.**title('With perplexity = 50, n\_iter=5000')

plt**.**show()

Chart, scatter chart

Description automatically generated

**IN [0]**

model **=** TSNE(n\_components**=**2, random\_state**=**0, perplexity**=**2)

tsne\_data **=** model**.**fit\_transform(data\_1000)

*# creating a new data fram which help us in ploting the result data*

tsne\_data **=** np**.**vstack((tsne\_data**.**T, labels\_1000))**.**T

tsne\_df **=** pd**.**DataFrame(data**=**tsne\_data, columns**=**("Dim\_1", "Dim\_2", "label"))

*# Ploting the result of tsne*

sn**.**FacetGrid(tsne\_df, hue**=**"label", size**=**6)**.**map(plt**.**scatter, 'Dim\_1', 'Dim\_2')**.**add\_legend()

plt**.**title('With perplexity = 2')

plt**.**show()

Chart, scatter chart

Description automatically generated

1. **Computer Viruses:**

**Identifying the problem:**

* Viruses spread when the software or documents they get attached to are transferred from one computer to another using a network, a disk, file sharing methods, or through infected e-mail attachments. Some viruses use different stealth strategies to avoid their detection from anti-virus software.
* One thing you'll notice that all of these infection vectors have in common is that they require the victim to execute the infected application or code. Remember, a virus can only execute and reproduce if its host application is running! Still, with email such a common malware dispersal method, a question that causes many people anxiety is: Can I get a virus from opening an email? The answer is that you almost certainly can't simply by opening a message; you have to download and execute an attachment that's been infected with virus code. That's why most security pros are so insistent that you be very careful about executing email attachments, and why most email clients and webmail services include virus scanning features by default.
* So, the problem clearly states a virus can only infect if it gets executed after bypassing all antiviruses and checks that computer do before executing a file.

**Solution:**

* Do not install third party programs that windows allow.
* Avoid installing those programs on which your PC is giving warning because those programs may contains harmful code.
* Check if your mails are coming from legit mail that you are intended to accept if not do a proper research if it contains a virus or not.
* If by mistakes virus get executed then search for its process in task manger and kill that process for stopping it.

1. **Unable To connect to Internet:**

**Identifying the problem:**

The main reason by which this problem arises are following:

1. Unplugged or loose network cables, routers, and modems are easy to miss. Yet, these devices are a common reason you might be unable to connect to the internet
2. What may seem to be a network problem connecting to the internet may be a website or server that is temporarily offline.
3. If your computer and another device on the network have the same IP address, the conflict between them prevents either from working properly online. To resolve an IP conflict, release and renew your IP address. If your network uses static IP addresses, manually change your IP to a different number.
4. Firewall software prevents unwanted network traffic from disrupting its operation. These software firewalls can malfunction and block valid internet traffic. When two software firewalls, such as Windows Firewall and a third-party product, are installed on the same computer, contention between the two can also incorrectly block traffic. If you recently installed or upgraded a software firewall on your computer, temporarily disable it to determine whether it may be the cause of your internet connection problem.
5. The performance of Wi-Fi network connections depends on the distance between the device and the wireless access point. The farther away from a Wi-Fi device, the slower the local connection. Wireless signal interference in the area can also limit the range of a Wi-Fi connection. If you can't reach the access point and can't connect to the internet, measure the strength of your wireless signal and then expand the range of your Wi-Fi.

**Solution:**

* Confirm your wired or wireless network hardware is switched on and plugged in.
* Before assuming your internet connection is faulty, visit several websites rather than just one.
* Avoid IP Address conflicts.
* Check for computer firewall malfunctions.
* Make sure you're within wireless signal range.

1. **Duplicate Ip Address:**

**Identifying the problem:**

1. Your system administrator could have assigned two computers on a local area network (LAN) the same static IP address.

2. Your internet service provider accidentally assigned two people the same IP address.

3. The network’s Dynamic Host Configuration Protocol (DHCP) server has allowed the same dynamic address to be assigned to multiple computers automatically.

4. Your system administrator has assigned a static IP address to a computer within the local network’s DHCP range, and the same address is automatically given by the local DHCP server.

**Solution**:

* Try changing your local Ip address either put it to automatic select the available Ip address mode or you can try putting some custom ip addresses to change the IP address.
* The second way to see the available IP address is by seeing all the IP addresses that are currently in use. You can check ip address which are already in use by using netdiscover command on Linux. And then after seeing all IP addresses you can choose your IP address which you don’t found in list.
* Contact your Internet Service Provider if the problem is from there. They may have assigned similar IP addresses to two networks by mistake.

1. **Slow Performance:**

**Identifying the problem:**

Slow-running applications can put a damper on your productivity in the workplace. One of the most common network issues that business networks fight with is slow applications. This happens especially when a computer first turns on or connects to a network. In most cases, this is caused by heavy bandwidth usage. In other instances, it can be caused by lack of hard drive space, running too many applications at once, having too many browser tabs open at one time, or even just a dusty room! The solution for this issue depends on the root of the problem.

Once you’ve gotten rid of some of your browser’s extensions, eliminated applications you aren’t using, or identified the application that’s eating up all of your processing power, you should be able to see a huge difference in your computer’s processing speed.

**Solution:**

* When you have so many application running in your pc then RAM of your computer may have too much load. What happens suppose your running applications is crosses size greater than the RAM size then computer will starts using the concept of virtual memory in which it start using the space which is available in Hard disk, making the computer speed slow because RAM is faster than Hard Disk. So, if you increase your RAM size. You program will stay in RAM hence making it faster.
* If you cannot increase your RAM then you can avoid using multiple big programs at same time keeping the RAM a bit free so that it can execute the background processes easily.
* Disable startup programs
* Scan for malware and viruses
* Upgrade your memory
* Install system updates