```
#!/usr/bin/env python
# coding: utf-8


# Speech-to-Text Import.
import speech_recognition as sr
# Text-to-Speech Import.
import gtts
from playsound import playsound
# To remove the punctuation and upper case letters from the user's pre-made note.
import string
# To build Graphical User Interfaces.
from tkinter import *
# Add a full-screen scrollbar as ttk widget.
from tkinter import ttk
# To change the font type of Tkinter button.
import tkinter.font as font
# A data structure used in manipulating the user's note.
from collections import defaultdict

# Create a root widget that corresponds to the main window.
root = Tk()
root.title("PROMPT ME OUT!")
root.iconbitmap("/Users/andersonyou/Desktop/Year 3/Individual Project/21-22/Final Prototype/teleprompter.ico")
root.geometry("800x600") # default size of the main window

# The section below before defining users_note variable refers to a general method for
# adding a full-screen scrollbar to the main window in Tkinter.
#
# Adapted from a YouTube tutorial video delivered by John Elder:
# URL: https://www.youtube.com/watch?v=0WafQCaok6g
#
# Create a main frame.
main_frame = Frame(root)
main_frame.pack(fill = BOTH, expand = 1)

# Create a canvas.
my_canvas = Canvas(main_frame)
my_canvas.pack(side = LEFT, fill = BOTH, expand = 1)

# Add a scrollbar to the canvas.
my_scrollbar = ttk.Scrollbar(main_frame, orient = VERTICAL, command = my_canvas.yview)
my_scrollbar.pack(side = RIGHT, fill = Y)

# Configure the canvas.
my_canvas.configure(yscrollcommand = my_scrollbar.set)
my_canvas.bind('<Configure>', lambda e: my_canvas.configure(scrollregion = my_canvas.bbox("all")))

# Create another frame inside the canvas.
second_frame = Frame(my_canvas)

# Add that new frame to a window in the canvas.
my_canvas.create_window((0,0), window = second_frame, anchor = "nw")

global users_note
users_note = """
                Hello, I am Anderson and I was working on a small piece of technology consulting project for EY.

                Recently, my company had been fined by the violation of Regulation 1215, which highlights that the number of people in their office wearing a face mask, and the number of people in their office at any point in time.
                I was told to find some solutions by achieving four objectives, which are desk bookings and allocations' prediction, secure storage of personal data, using visualization tools for vendor comparison and monitoring social distancing.

                The first technology I am going to talk about refers to Machine Learning, and precisely Supervised Learning, which can be used to effectively forecast desk bookings and allocations. So what is Supervised Learning?
                Well if you have a look at the diagram on the top right, let's assume that the Model is a Mathematical function say y = 3x, and assign the training input data to be 2, 3 and 4, then every time we feed those inputs into the model,
                it will get updated and output 6, 9 and 12 respectively. Now, this model has got trained and that means if we give it a brand-new piece of data, we can check whether the system works as intended. So 5 outputs 15, 6 outputs 18,
                and so on. In this example, we can use one of the Supervised Learning techniques called Regression, which consists of the horizontal axis representing the distance to café/canteen in meters, and
```

the vertical axis representing
the length since they first joined EY. As we assume that more senior people tend to book more interior places so much less noise and distraction, and for those who newly joined will choose more public working spaces in order to
network with more people and get familiarized within the company. We can collect some data by randomly picking 10 people, plotting the data point, and drawing a line of best fit. The predictions can then be shown by
further extrapolating the line. The line of best fit can also be obtained by calculation, but I won't go through them in detail, just for the sake of interest.

The second technology is more like a combination of Cyber Security and Cloud Computing in order to securely store personal data relating to desk bookings. Cryptography, it's one of the specific parts of Cyber Security,
which manipulates data for the purpose of hiding and authenticating information. For Cloud Computing, it's on-demand access, via the internet, to computing resources (applications, servers, development tools etc). This is hosted
at a remote data centre managed by a cloud services provider (or CSP in short). In this case, all personal data can be encrypted into a cloud using the Symmetric Cryptographic Algorithm. The image at the bottom left is
a simple example showing how the plaintext is converted into ciphertext using some encryption function. And the one on the right-hand side gives more intuition. Basically, the key here means that the encryption algorithm
can be derived from the decryption algorithm and vice versa. The plaintext and the ciphertext both have the same size. And for ciphertext, it's converted specifically via a Block Cipher, which takes one block of data at a time
for each conversion. When using these kinds of technology, there are four things that get enforced for storing data. The first one is Confidentiality, which ensures that others without the key cannot read the content of the data.
The second one is Integrity, which verifies that the data has not been modified. Then for Authentication, it determines where this data has come from, and finally, Nonrepudiation ensures that the sender (in this case people
who put all the information into the cloud) should not be able to falsely deny that the data was sent.

The third technology refers to Artificial Intelligence, what I mean in this specific example is the symbolic/classical AI approach, which is a methodology or process derived from a human's written code, and from that,
the system can search for good strategies to solve problems within the solutions' domain. Here, I am using a video-based people-tracking software deployed in Miami airport, which I think is also suitable for EY. Let's imagine
every person in there is a moving dot. You can get a really good kind of image from above of where everyone is, how they cluster, how they're moving, how far apart they are from each other, and so on. The camera on the ceiling
then uses the data to generate a "score". You can see green, red and yellow here, pretty much like a traffic light on the road, and green are the people that we're maintaining a social distance of six feet or more.
Audio announcements will be generated if too many people are not obeying the social-distancing rules.

Now moving on to the Visualization Tool comparison for the vendor, there are around 20 data visualization tools by quick Googling, and I will pick three of them, which is Microsoft Power BI, Tableau and Google Charts
for the sake of this short presentation. So let's compare them by walking through each of their characteristics: Firstly, almost all tools are premise/cloud-based except for Google Charts. Tableau and Google Charts don't really
have limits for direct data upload while there's a 1GB limit per dataset for Microsoft Power BI. For data cleansing tools, neither Microsoft Power BI nor Google Charts has one whereas Tableau has a Tableau Prep Software
which contains data cleaning functionality. And all three tools are capable of drilling down data as well as exporting them to Excel and PowerPoint. Regarding mobile compatibility, Microsoft Power BI and Google Charts
are both compatible with Android and iOS, while Tableau is still under deployment. They can all be integrated with Information systems, specifically for Microsoft Power BI with OutSystems, Tableau with Tealium and
Google Charts with ASP.NET Webform. And finally, for integration with programming software, Microsoft Power BI with Tidal Software, Tableau with TabPy for Python Integration, whereas Google Chart doesn't integrate with
any of the programming software.

Putting them all together, the solutions of those four objectives mentioned on the second slide has all got clarified. So use linear regression to predict desk bookings and allocation, use Symmetric Cryptographic Algorithm
in conjunction with Cloud Computing to securely store personal data, use Microsoft Power BI, Tableau and Google Charts as three visualization tools for vendor comparison, and finally use the video-based smart camera to
monitor social distancing. In addition to the advantage of the three technologies I mentioned earlier on the slides, using visualization tools allows businesses to improve their ability to extract relevant insights from
within large datasets, as well as quickly help them identify relevant patterns and trends hidden in the data. So are there any risks/costs that exist? Yes and that's what the next slide is going to be.

For the first technology, what if there's some data point lies too far away from the line of best fit? This introduces what we called bias, and the way to fix this is to try and draw a curve

```
instead of a straight line in order to
99                bring the points closer to the line of best fit, which is called Curvilinear Regression.
For the second technology, let's consider a scenario that the managers and every employee have access to
the cloud and are able to know
100               where everyone sits within the office. The problem is that some people may not want others
to know where he/she exactly is during working hours for the sake of less distraction. That's when
Asymmetric Cryptographic Algorithm
101               comes into place (the diagram pretty much shows how it works), which ensures that everyone
can get a distinct private key so that when they log into the cloud, they can only see his/her booking, and
the public key can then
102               be handed into a more senior team (say the people who monitor the entire reservation
process). And for the third one, the camera in some cases may misinterpret the social-distancing rules, for
instance when a group of people
103               walking close to each other but they're actually a group of family or friends. This is the
problem that the company itself should try to figure out and do some optimization. And the last thing, for
using visualization tools,
104               the people using it may not have enough knowledge beforehand regarding the business
organization, data and their corresponding definition. It generally takes a long time for people who get
trained by developing all these skills.
105
106               That's the end of the video, thanks for watching!
107            """
108
109 global note_label
110 # Create a Label widget to hold the user's pre-made note with left align.
111 note_label = Label(second_frame, text = users_note, justify = "left")
112 # Shove it onto the screen, with some padding below the border.
113 note_label.grid(row = 0, column = 0, pady = 15)
114
115 output_label = Label(second_frame)
116
117 def prompter(current_index = defaultdict(int)):
118
119     """
120     Prompts the cue word right next to the input word via text and sound.
121
122     Parameters
123     ----------
124     current_index : collections.defaultdict
125         This is bound to a new defaultdict(int) only once, when the function is defined here,
126         not each time the function is called. Therefore current_index is preserved between calls.
127         Key = word entered (e.g. "you")
128         Value = index of word's previous next word (e.g. "2" defaults to 0)
129
130     Raises
131     ------
132     ValueError
133         If the word entered either has no more occurrences for the rest of the text,
134         or it cannot be found inside the text.
135     """
136
137     # Convert the string format of text into a list of strings.
138     note_list = users_note.split()
139     # Remove punctuation and upper case letters for the system's output.
140     note_list = [''.join(letter.lower() for letter in word if letter not in string.punctuation) for word in
note_list]
141
142     # Initialize the recognizer.
143     r = sr.Recognizer()
144     with sr.Microphone() as source:
145         # Read the audio data from the default microphone, give user 8 seconds to respond.
146         audio_data = r.record(source, duration = 8)
147         # Convert speech to text.
148         word = r.recognize_google(audio_data)
149
150         # Get the index of the word, starting from the word's previous
151         # next word index, which is stored in current_index[word], then increment 1.
152         try:
153             next_word_index = note_list.index(word, current_index[word]) + 1
154
155         # There are either no more occurrences of the word, or the word doesn't exist at all.
156         except ValueError:
157             if word in note_list:
158                 last_occurrence = f'There are no more occurrences of {word} in the text, please try again'
159                 global output_label
160                 # Delete the previous system's output.
161                 output_label.destroy()
162                 # Update the system's output.
163                 output_label = Label(second_frame, text = last_occurrence, font = ("Courier", 15))
```

```
                # Locate the system's output right below the button.
                output_label.grid(row = 3, column = 0, pady = 30)
                # Make request to Google to get synthesis.
                tts = gtts.gTTS(last_occurrence)
                # Save the audio file.
                tts.save("ValueError1.mp3")
                # Play the audio file.
                playsound("ValueError1.mp3")
            else:
                output_label.destroy()
                output_label = Label(second_frame, text = "Sorry, try again", font = ("Courier", 15))
                output_label.grid(row = 3, column = 0, pady = 30)
                tts2 = gtts.gTTS('Sorry, try again')
                tts2.save("ValueError2.mp3")
                playsound("ValueError2.mp3")
            return

        # Update the word's current index.
        current_index[word] = next_word_index

        # When the input word is the last word of the user's note.
        if next_word_index == len(note_list):
            output_label.destroy()
            output_label = Label(second_frame, text = "This is the last word of the text, please try
again", font = ("Courier", 15))
            output_label.grid(row = 3, column = 0, pady = 30)
            tts3 = gtts.gTTS('This is the last word of the text, please try again')
            tts3.save("IndexOutOfBoundError.mp3")
            playsound("IndexOutOfBoundError.mp3")
            return

        # Yield the next word.
        next_word = note_list[next_word_index]
        output_label.destroy()
        output_label = Label(second_frame, text = "The next word is: " + next_word, font = ("Courier", 15))
        output_label.grid(row = 3, column = 0, pady = 30)
        tts4 = gtts.gTTS(next_word)
        tts4.save("output.mp3")
        playsound("output.mp3")

def main():

    """
    Calls the previous function and handles another exception.

    Raises
    ------
    UnknownValueError
        If no input has been detected by the system for more than 8 seconds.
    """

    try:
        prompter()

    # When the user doesn't produce any response for more than 8 seconds.
    except sr.UnknownValueError:
        global output_label
        output_label.destroy()
        output_label = Label(second_frame, text = "Time limit exceeded, please try again", font =
("Courier", 15))
        output_label.grid(row = 3, column = 0, pady = 30)
        tts5 = gtts.gTTS('Time limit exceeded, please try again')
        tts5.save("UnknownValueError.mp3")
        playsound("UnknownValueError.mp3")
        return

# Create a Button widget that enables users to press when they get stuck on a word, customise the font.
buttonFont = font.Font(family = 'Helvetica', size = 16, weight = 'bold')
btn = Button(second_frame, text = "Tell me the last word before you get stuck: ", font = buttonFont,
command = main)
# Locate the button right below the user's note.
btn.grid(row = 1, column = 0)

# Add some space between the button and the system's output.
Label(second_frame, text = "\n").grid(row = 2, column = 0)

# Centre all the contents displayed in the window.
```

```
second_frame.columnconfigure(0, weight = 1)
second_frame.rowconfigure(0, weight = 1)

# Create an event loop.
root.mainloop()


# Useful Links:
# https://www.thepythoncode.com/article/using-speech-recognition-to-convert-speech-to-text-python
# https://www.thepythoncode.com/article/convert-text-to-speech-in-python
# https://www.techbeamers.com/python-multiline-string/
# https://stackoverflow.com/questions/17686809/how-to-find-word-next-to-a-word-in-python
# https://www.programcreek.com/python/example/107723/speech_recognition.UnknownValueError
# https://www.delftstack.com/howto/python/python-remove-punctuation-from-list/
# https://stackoverflow.com/questions/55343738/how-do-you-use-tkinter-to-display-the-output-of-a-function-call
# https://stackoverflow.com/questions/42828416/print-output-in-gui-interface-tkinter-python
# https://www.tutorialspoint.com/python/tk_fonts.htm
# https://www.tutorialspoint.com/python/tk_text.htm
# https://stackoverflow.com/questions/40237671/python-tkinter-single-label-with-bold-and-normal-text
# https://stackoverflow.com/questions/46069531/python-how-to-center-label-in-tkinter-window
# https://stackoverflow.com/questions/10851906/python-3-unboundlocalerror-local-variable-referenced-before-assignment
# https://www.geeksforgeeks.org/defaultdict-in-python/
# https://www.geeksforgeeks.org/python-list-index/
# https://www.tutorialkart.com/python/tkinter/button/font/
# https://www.geeksforgeeks.org/how-to-change-the-tkinter-label-font-size/
# https://www.delftstack.com/howto/python-tkinter/how-to-set-font-of-tkinter-text-widget/#:~:text=Set%20Font%20for%20Tkinter%20Text%20Widget%20With%20tkFont,-We%20could%20also&text=family%20%2D%20font%20family%2C%20like%20Arial,font%20slant%3A%20roman%20or%20italic
# https://www.tutorialspoint.com/how-to-justify-text-in-label-in-tkinter-in-python-need-justify-in-tkinter
# https://www.youtube.com/watch?v=0WafQCaok6g
# https://www.youtube.com/watch?v=Q-rRF6c8kJM&t=36s
# https://www.flaticon.com/free-icons/teleprompter?word=teleprompter&type=icon&order_by=4
# https://stackoverflow.com/questions/48981184/set-window-icon-tkinter-macosx
# https://realpython.com/documenting-python-code/
# https://realpython.com/python-main-function/
```