In [1]:

```
import tkinter
from tkinter import ttk
from tkinter import *
from tkinter import filedialog
import pandas as pd
```

In [2]:

```
homeScreen = Tk()
homeScreen.title("Great Learning - Capstone 1")
homeScreen.geometry("400x500")
homeScreen.resizable(width=TRUE, height=TRUE)

tabControl = ttk.Notebook(homeScreen)
```

In [4]:

```
def browseFiles():
    File_Entry.delete('0', END)
    filename = filedialog.askopenfilename(initialdir = "/",
                                          title = "Select a File",
                                          filetypes = (("CSV files",
                                                        "*.csv*"),
                                                       ("all files",
                                                        "*.*")))
    File_Entry.insert(END,filename)
```

In [5]:

```
def preprocessDataset():
    filename = File_Entry.get()
    data = pd.read_csv(filename)
    writeToConsole("CSV File Read")
    data = preprocessFile.preprocess_dataset(data)
    writeToConsole("Basic Data Processing Done")
    data = preprocessFile.stopwords_removal(data)
    writeToConsole("Stop Words Removed")
    data = preprocessFile.data_lemmatization(data)
    writeToConsole("Lemmatization Completed")
    data = preprocessFile.spell_checking(data)
    writeToConsole("Spelling Check Done")
    data = preprocessFile.frequent_words(data)
    writeToConsole("Frequent Words Extracted")
    ## Output Dataset
    fileNameModified = filename.split('/')[-1]
    fileNameModified = fileNameModified.split('.')[0]
    writeToConsole("Writing Processed Data to CSV File")
    data.to_csv(f'Output_{fileNameModified}.csv')
    writeToConsole("CSV file downloaded")
```

In [6]:

```
def writeToConsole(logmsg):
    Console.config(state=NORMAL)
    Console.insert(END, logmsg + '\n')
    Console.config(foreground="black", font=("Cambria", 12 ))
    Console.config(state=DISABLED)
    Console.yview(END)
```

In [ ]:

```
#function called is present in another module which has same source code as used
in the main ipynb filed shared
def preprocess():
    txt = EntryBox_Preprocess.get("1.0",'end-1c').strip()
    if txt != '':
        preprocessedSentence = pre.pre_processing_text(txt)
        Box_Processed.config(state=NORMAL)
        Box_Processed.delete("0.0",END)
        Box_Processed.insert(END,preprocessedSentence )
        Box_Processed.config(foreground="red", font=("Cambria", 12,'bold' ))
```

In [ ]:

```
def browseOutputFiles():
    File_Entry_1.delete('0', END)
    filename_1 = filedialog.askopenfilename(initialdir = "/",
                                            title = "Select a File",
                                            filetypes = (("CSV files",
                                                          "*.csv*"),
                                                         ("all files",
                                                          "*.*")))
    File_Entry_1.insert(END,filename_1)
```

In [ ]:

```
#function called is present in another module which has same source code as used
in the main ipynb filed shared

def train_model_al():
    filename_1 = File_Entry_1.get()
    data = pd.read_csv(filename_1)
    x_train,x_test,y_train,y_test,labels_al, num_words,embedding_size, embedding
_matrix, maxlen, Y_AL = al_train.prepare_X_Y(data)
    writeToConsole2("AL Test Train Split Done")
    model = al_train.build_model(x_train,x_test,y_train,y_test,labels_al, num_wo
rds,embedding_size, embedding_matrix, maxlen, Y_AL)
    writeToConsole2("AL Model Build")
    al_train.model_save(model)
    writeToConsole2("AL Model Stored to Disk")

def train_model_pal():
    writeToConsole2("Training PAL Model")
    filename_1 = File_Entry_1.get()
    data = pd.read_csv(filename_1)
    x_train_p,x_test_p,y_train_p,y_test_p,labels_pal, num_words,embedding_size,
embedding_matrix, maxlen, Y_PAL = pal_train.prepare_X_Y(data)
    writeToConsole2("PAL Test Train Split Done")
    model = pal_train.build_model(x_train_p,x_test_p,y_train_p,y_test_p,labels_p
al, num_words,embedding_size, embedding_matrix, maxlen, Y_PAL)
    print("PAL Model Build")
    pal_train.model_save(model)
    writeToConsole2("PAL Model Stored to Disk")
```

In [ ]:

```
#function called is present in another module which has same source code as used
in the main ipynb filed shared

def predict_al(textString, dataset):
    preprocessedtextString = pre.pre_processing_text(textString)
    print(preprocessedtextString)
    X_input = pmodel.tokenize(preprocessedtextString)
    print("Tokenized")
    X_padded = pmodel.sequence_padding(X_input)
    print("Sequence Padded")
    #Y = pmodel.label_binarize(data)
    acc_level  = pred.predict_al(X_padded, dataset)
    print(acc_level)
    return acc_level

def predict_pal(textString, dataset):
    preprocessedtextString = pre.pre_processing_text(textString)
    print(preprocessedtextString)
    X_input = pmodel.tokenize(preprocessedtextString)
    print("Tokenized")
    X_padded = pmodel.sequence_padding(X_input)
    print("Sequence Padded")
    #Y = pmodel.label_binarize(data)
    potential_acc_level  = pred.predict_pal(X_padded, dataset)
    print(potential_acc_level)
    return potential_acc_level
```

In [ ]:

```
def get_output_file():
    dataset = pd.read_csv(filename_1)
    return dataset

#function called is present in another module which has same source code as used
in the main ipynb filed shared
def prediction(textString):
    print(textString)
    dataset = get_output_file()
    al_prediction = predict_al(textString, dataset)
    pal_prediction = predict_pal(textString, dataset)
    response = "AL: " + al_prediction + '\n' + "PAL: " + pal_prediction + '\n\n'
    return response
```

In [ ]:

```
#function called is present in another module which has same source code as used
in the main ipynb filed shared
def send():
    msg = EntryBox.get("1.0",'end-1c').strip()
    EntryBox.delete("0.0",END)

    if msg != '':
        ChatBox.config(state=NORMAL)
        ChatBox.insert(END, "Me: " + msg + '\n\n')
        ChatBox.config(foreground="red", font=("Cambria", 12,'bold' ))

        response = prediction(msg)
        ChatBox.insert(END,response)
        ChatBox.config(foreground="red", font=("Cambria", 12,'bold' ))

        ChatBox.config(state=DISABLED)
        ChatBox.yview(END)
```

In [3]:

```
# First Tab
tab_preprocess = ttk.Frame(tabControl)
tabControl.add(tab_preprocess, text ='Pre-Processing')
tabControl.pack(expand = 1, fill ="both")

EntryBox_Preprocess = Text(tab_preprocess, bd=4, bg="#E1F6F9",width="33", height
="5", font="Cambria")

SendButton_Preprocess = Button(tab_preprocess, font=("Arial Black",10,'bold'), t
ext="Test Data Preprocessing", width="12", height=5,
bd=0, bg="#9F0404", activebackground="#FF5733",fg='#F2EBEB',command=preprocess)

Box_Processed = Text(tab_preprocess, bd=4, bg="#E1F6F9",width="33", height="5",
font="Cambria")

Console = Text(tab_preprocess, bd=4, bg="#FFFAFA",width="33", height="5", font=(
"Cambria",8))
Console.config(state=DISABLED)

SendButton_DatasetPreprocess = Button(tab_preprocess, font=("Arial Black",10,'bo
ld'), text="Pre - process Dataset", width="12", height=5,
bd=0, bg="#9F0404", activebackground="#FF5733",fg='#F2EBEB',command= preprocessD
ataset)
x = StringVar()

File_Entry = Entry(tab_preprocess,textvariable=x,justify='center',bd=2)

Browse_button = Button(tab_preprocess,text = "Browse File", fg="red",command=bro
wseFiles)
```

In [ ]:

```
# Second Tab
tab_model = ttk.Frame(tabControl)
tabControl.add(tab_model, text ='Model Training')
tabControl.pack(expand = 1, fill ="both")
tab_chat = ttk.Frame(tabControl)

File_Entry_1 = Entry(tab_model,textvariable=x,justify='center',bd=2)
Browse_output_button = Button(tab_model,text = "Browse Output File", fg="red",co
mmand=browseOutputFiles)

Button_ALModel = Button(tab_model, font=("Arial Black",10,'bold'), text="Model T
raining For AL", width="12", height=5,
bd=0, bg="#9F0404", activebackground="#FF5733",fg='#F2EBEB',command=train_model_
al )

Button_PALModel = Button(tab_model, font=("Arial Black",10,'bold'), text="Model
 Training For PAL", width="12", height=5,
bd=0, bg="#9F0404", activebackground="#FF5733",fg='#F2EBEB',command=train_model_
pal )

Console2 = Text(tab_model, bd=4, bg="#FFFAFA",width="33", height="5", font=("Cam
bria",8))
Console2.config(state=DISABLED)
```

In [ ]:

```
# Third Tab
tabControl.add(tab_chat, text ='Chatbot')

#Create Chat window
ChatBox = Text(tab_chat, bd=0, bg="white", height="8", width="50")

ChatBox.config(state=DISABLED)

#Bind scrollbar to Chat window
scrollbar = Scrollbar(tab_chat, command=ChatBox.yview, cursor="heart")
ChatBox['yscrollcommand'] = scrollbar.set

#Create Button to send message
SendButton = Button(tab_chat, font=("Arial Black",10,'bold'), text="Send", width
="12", height=5,
bd=0, bg="#9F0404", activebackground="#FF5733",fg='#F2EBEB',command= send
)

#Create the box to enter message
EntryBox = Text(tab_chat, bd=4, bg="#E1F6F9",width="33", height="5", font="Cambr
ia")
#EntryBox.bind("<Return>", send)
```

In [ ]:

```
# First tab placements
Console.place(x=6,y=100, height=100, width=384)
EntryBox_Preprocess.place(x=6,y=220, height=100, width=384)
Box_Processed.place(x=6, y=370, height=100, width=384)
File_Entry.place(x=6,y=6,height=30,width=300)
Browse_button.place(x=310,y=6,height=30,width=80)
SendButton_Preprocess.place(x=110, y=330, height=30, width = 200)
SendButton_DatasetPreprocess.place(x=110, y=50, height=30, width=200)
```

In [ ]:

```
# Second tab placements
File_Entry_1.place(x=6,y=6,height=30,width=370)
Browse_output_button.place(x=110,y=50,height=30,width=200)
Button_ALModel.place(x=6, y=140, height=60, width=180)
Button_PALModel.place(x=200, y=140, height=60, width=180)
Console2.place(x=6,y=250, height=250, width=384)
```

In [ ]:

```
#Third Tab Placements
scrollbar.place(x=376,y=6, height=386)
ChatBox.place(x=6,y=6, height=386, width=370)
EntryBox.place(x=6, y=401, height=80, width=269)
SendButton.place(x=276, y=401, height=84)

homeScreen.mainloop()
```