In [1]:

```
import nltk
import string
import requests
import json
```

In [2]:

```
import numpy as np
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import LinearSVC  #importing machine learning classification algorithm
import random
import pickle
import re
```

In [3]:

```
import telebot
import time
from telebot import types

import pyqrcode
import png
from pyqrcode import QRCode
```

In [ ]:

```

```

In [4]:

```
# training data
data = [
    ['Hello',0],
    ['Hey',0],
    ['Hi',0],
    ['Good Morning',0],

    ['I would like to book a slot',1],
    ['I would like to do a booking',1],
    ['I want to book a slot',1],
    ['Book a slot for me',1],
    ['Book a slot',1],
    ['Want a booking',1],
    ['Want to do a booking',1],
    ['Want a slot',1],
    ['new appointment',1],

    ['what is the cost of the booking',2],
    ['what is the cost',2],
    ['can i see the menu',2],
    ['menu',2],
    ['i would like to see the menu',2],
    ['i would like to see the catalog',2],
    ['catalog',2],

    ['timeslots',3],
    ['timings',3],
    ['time',3],
    ['i want to select the time slot',3],

    ['confirmation',4],
    ['confirm',4],

    ['thanks',5],
    ['thank',5],
    ['thanks you',5],
    ['bye',5],
    ['No thank you',5],

    ['cancel',6],
    ['dont book',6],



]
```

['10 AM',10], ['11 AM',10], ['12 AM',10], ['1 PM',10], ['2 PM',10], ['3 PM',10], ['4 PM',10], ['5 PM',10], ['6 PM',10], ['7 PM',10],

```
['Haircut, Price: 80',10],
['Shave, Price: 100',10],
['Hair Spa, Price: 150',10],
['Head Massage, Price: 150',10],
['Hair Styling, Price: 150',10],
['Hair Coloring, Price: 200',10],
['Shave + Haircut, Price: 150',10],
['Head Massage + Shave, Price: 200',10]
```

In [ ]:

In [ ]:

In [ ]:

In [5]:

```
#converting it into pandas dataframe
df = pd.DataFrame(data, columns = ['text', 'intent'])
```

In [6]:

```
df   #1 -> availability and 0 -> greeting
```

Out[6]:

| | text | intent |
|---|---|---|
| **0** | Hello | 0 |
| **1** | Hey | 0 |
| **2** | Hi | 0 |
| **3** | Good Morning | 0 |
| **4** | I would like to book a slot | 1 |
| **5** | I would like to do a booking | 1 |
| **6** | I want to book a slot | 1 |
| **7** | Book a slot for me | 1 |
| **8** | Book a slot | 1 |
| **9** | Want a booking | 1 |
| **10** | Want to do a booking | 1 |
| **11** | Want a slot | 1 |
| **12** | new appointment | 1 |
| **13** | what is the cost of the booking | 2 |
| **14** | what is the cost | 2 |
| **15** | can i see the menu | 2 |
| **16** | menu | 2 |
| **17** | i would like to see the menu | 2 |
| **18** | i would like to see the catalog | 2 |
| **19** | catalog | 2 |
| **20** | timeslots | 3 |
| **21** | timings | 3 |
| **22** | time | 3 |
| **23** | i want to select the time slot | 3 |
| **24** | confirmation | 4 |
| **25** | confirm | 4 |
| **26** | thanks | 5 |
| **27** | thank | 5 |
| **28** | thanks you | 5 |
| **29** | bye | 5 |
| **30** | No thank you | 5 |
| **31** | cancel | 6 |
| **32** | dont book | 6 |

In [7]:

```
x = df['text']   #seperating x from data
y = df['intent']   # seperating y from data (intents are encoded into numbers as machin
e can only predict numbers)
```

In [8]:

```
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(x)  # X has tfidfed vectors
```

In [9]:

```
clf = LinearSVC(max_iter=800,C=0.1)
```

In [10]:

```
clf.fit(X,y) #creating the model. after fitting model is ready
```

Out[10]:

```
LinearSVC(C=0.1, class_weight=None, dual=True, fit_intercept=True,
          intercept_scaling=1, loss='squared_hinge', max_iter=800,
          multi_class='ovr', penalty='l2', random_state=None, tol=0.0001,
          verbose=0)
```

In [11]:

```
pickle.dump(clf,open("model.pkl", 'wb')) #saving
```

In [12]:

```
load_model = pickle.load(open("model.pkl", 'rb')) #loading
```

In [13]:

```
responses = {0 : {"intent":"greetings",
                  "response":["Hi dear, Wellcome to 'BigMan Saloon'. How can i help yo
u.", "How are you? Wellcome to 'BigMan Saloon'. How can i help you.", "Hello! Wellcome
 to 'BigMan Saloon'. How can i help you."]},

             1 : {"intent":"availability",
                  "response":["Yes Yes it is available (type 'Menu')","Yes seats are av
ailable(type 'Menu')","Ofcourse there are seats for you(type 'Menu')"]},

             2: {"intent":"catalog",
                 "response":['I will show you the menu ' , 'Yes i will show you the cat
alog', 'I will show you the pricing']},

             3: {"intent":"timing",
                 "response":[ "We have provided flexible timeslots. "]},

             4: {"intent":"confirmation",
                 "response":[ "Your confirmation is here. "]},

             5: {"intent":"thanks",
                 "response":[ "Payment needs to be done at the solon. Online payment at
the salon can be done. \nHappy to help 😊"]},

             6: {"intent":"cancel",
                 "response":[ "Your your booking has been cancelled. ", "Dont worry you
will not be charged , we have cancelled your booking"]}


            }
```

In [14]:

```
global pd
def make_reply(user_response):
    text_test = [user_response]
    X_test = vectorizer.transform(text_test)
    prediction = clf.predict(X_test)
    pd = prediction
    reply = random.choice(responses[prediction[0]]["response"])
    return reply
```

In [ ]:

```

```

In [15]:

```
def hasNumbers(inputString):
    return any(char.isdigit() for char in inputString)
```

In [16]:

```
user_catalog_choice = ""
user_timing_choice = ""
user_date_choice = ""
```

In [17]:

```
bot_tokken = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" #your telegram bot tokken
bot = telebot.TeleBot(bot_tokken)
```

In [18]:

```
initial_msg = """Hello! This is a chatbot and my name is Lucas. I will guide you throug
h the booking process.\nAs you know due to this current pandemic situation people are a
voiding salons to dont get in contact with the virus.
\nSo here i offer you a online booking system , so there is no need of any hassle or wa
iting in queue.
\nAt the given booking time only one person will be handeled and we assure you totally
 sanitized and good hygenic environment. /help."""

help_msg = "How may i help you ?"
```

In [19]:

```
@bot.message_handler(commands=['start'])
def send_welcome(message):
    bot.reply_to(message, initial_msg)

    name = message.from_user.first_name
    msg = message.text
    user_id = message.from_user.id

    print("user name : " + name)
    print("user message : " + msg)
    print("robo message : " + initial_msg)
```

In [20]:

```
@bot.message_handler(commands=['help'])
def send_welcome(message):
    bot.reply_to(message, help_msg)

    name = message.from_user.first_name
    msg = message.text
    user_id = message.from_user.id

    print("\nuser name : " + name)
    print("user message : " + msg)
    print("robo message : " + help_msg)
```

In [21]:

```
@bot.message_handler(func=lambda message: True)
def echo_all(message):

    name = message.from_user.first_name
    msg = message.text
    user_id = message.from_user.id

    if(msg.find("confirm") != -1 or msg.find("Confirm") != -1 or msg.find("Confirmatio
n") != -1 or msg.find("confirmation") != -1 ):
        reply = make_reply(message.text)
        bot.send_message(message.chat.id, reply)
        markup = types.ReplyKeyboardMarkup(row_width=1)
        itembtn1 = types.KeyboardButton('YES')
        itembtn2 = types.KeyboardButton("NO")
        markup.add(itembtn1, itembtn2)
        bot.send_message(message.chat.id, "Please confirm :", reply_markup=markup)
        print("\nuser name : " + name)
        print("user message : " + msg)
        print("robo message : " + "Please confirm.")


    elif (message.text == "YES"):
        bot.send_message(message.chat.id, "Please while ticket is being generated.")

        user_info = "Name: " + name + "\nId: " + str(user_id)
        qrcode = pyqrcode.create(user_info)
        qrcode.png('myqr.png', scale = 6)
        photo = open('myqr.png', 'rb')

        bot.send_photo(message.chat.id, photo)
        bot.send_message(message.chat.id, "Here is the exact location of the salon.")
        bot.send_location(message.chat.id, 28.607450, 77.220420)
        bot.send_message(message.chat.id, "Please try be on time so as to avoid any pro
blem.")
        markup = types.ForceReply(selective=False)
        bot.send_message(message.chat.id, "Want to ask anything else.", reply_markup=ma
rkup)

        bot.send_message(message.chat.id, "Here is the exact location of the salon.")

        print("\nuser name : " + name)
        print("user message : " + msg)
        print("robo message : " + "Please while ticket is being generated.")


    elif (message.text == "NO" ):
        markup = types.ForceReply(selective=False)
        bot.send_message(message.chat.id, "Your seat will be cancelled.", reply_markup=
markup)
        print("\nuser name : " + name)
        print("user message : " + msg)
        print("robo message : " + "Your seat will be cancelled.")


    else:
        reply = make_reply(message.text)


        #if (msg.find("timing") != -1 or msg.find("timings") != -1 or msg.find("Timing
```

```
s") != -1 or msg.find("Timing") != -1 or reply.find("Time") != -1 ):
        if (np.where(pd == 3) and (msg.find("timing") != -1 or msg.find("timings") != -
1 or msg.find("Timing") != -1 or msg.find("time") != -1 or msg.find("Time") != -1)):
            bot.send_message(message.chat.id, reply)

            markup = types.ReplyKeyboardMarkup(row_width=2)
            itembtn1 = types.KeyboardButton("10 AM")
            itembtn2 = types.KeyboardButton("11 AM")
            itembtn3 = types.KeyboardButton("12 AM")
            itembtn4 = types.KeyboardButton("1 PM")
            itembtn5 = types.KeyboardButton("2 PM")
            itembtn6 = types.KeyboardButton("3 PM")
            itembtn7 = types.KeyboardButton("4 PM")
            itembtn8 = types.KeyboardButton("5 PM")
            itembtn9 = types.KeyboardButton("6 PM")
            itembtn10 = types.KeyboardButton("7 PM")
            markup.add(itembtn1, itembtn2,itembtn3, itembtn4, itembtn5, itembtn6, itemb
tn7, itembtn8, itembtn9, itembtn10)

            bot.send_message(message.chat.id, "TIMING", reply_markup=markup)

            print("\nuser name : " + name)
            print("user message : " + msg)
            print("robo message : " + "Timing")


        elif (msg.find("10 AM") != -1 or msg.find("11 AM") != -1 or msg.find("12 AM") !
= -1 or msg.find("1 PM") != -1 or msg.find("2 PM") != -1 or msg.find("3 PM") != -1 or m
sg.find("4 PM") != -1 or msg.find("5 PM") != -1 or msg.find("6 PM") != -1 or msg.find(
"7 PM") != -1):
            user_timing_choice = msg
            bot.send_message(message.chat.id, "Now lets choose date (type 'Choose dat
e')")

        elif (msg.find("date") != -1 or msg.find("Date") != -1):
            markup = types.ReplyKeyboardMarkup(row_width=2)
            itembtn1 = types.KeyboardButton("20")
            itembtn2 = types.KeyboardButton("21")
            itembtn3 = types.KeyboardButton("22")
            itembtn4 = types.KeyboardButton("23")
            itembtn5 = types.KeyboardButton("24")
            itembtn6 = types.KeyboardButton("25")
            itembtn7 = types.KeyboardButton("26")
            itembtn8 = types.KeyboardButton("27")
            markup.add(itembtn1, itembtn2,itembtn3, itembtn4, itembtn5, itembtn6, itemb
tn7, itembtn8)

            bot.send_message(message.chat.id, "DATE : Please choose the option you lik
e", reply_markup=markup)

            print("\nuser name : " + name)
            print("user message : " + msg)
            print("robo message : " + "Date")

        elif (msg.find("20") != -1 or msg.find("21") != -1 or msg.find("22") != -1 or m
sg.find("23") != -1 or msg.find("24") != -1 or msg.find("25") != -1 or msg.find("26") !
= -1 or msg.find("27") != -1  ):
            user_date_choice = msg
            bot.send_message(message.chat.id, "your name , booking date and timing has
 been recorded, would you like to confirm your booking (type confirmation)")
```

```
            bot.send_message(message.chat.id, "Now give your confirmation (type 'confir
m')")



        #elif (msg.find("Catalog") != -1 or msg.find("Menu") != -1 or msg.find("Pricin
g") != -1 ):
        elif (np.where(pd == 1) and (msg.find("Catalog") != -1 or msg.find("Menu") != -
1 )):
            bot.send_message(message.chat.id, reply)

            markup = types.ReplyKeyboardMarkup(row_width=2)
            itembtn1 = types.KeyboardButton("Haircut, Price: 80")
            itembtn2 = types.KeyboardButton("Shave, Price: 100")
            itembtn3 = types.KeyboardButton("Hair Spa, Price: 150")
            itembtn4 = types.KeyboardButton("Head Massage, Price: 150")
            itembtn5 = types.KeyboardButton("Hair Styling, Price: 150")
            itembtn6 = types.KeyboardButton("Hair Coloring, Price: 200")
            itembtn7 = types.KeyboardButton("Shave + Haircut, Price: 150")
            itembtn8 = types.KeyboardButton("Head Massage + Shave, Price: 200")
            markup.add(itembtn1, itembtn2,itembtn3, itembtn4, itembtn5, itembtn6, itemb
tn7, itembtn8)

            bot.send_message(message.chat.id, "CATALOG : Please choose the option you l
ike", reply_markup=markup)

            print("\nuser name : " + name)
            print("user message : " + msg)
            print("robo message : " + "Catalog")


        elif (msg.find("Haircut, Price: 80") != -1 or msg.find("Shave, Price: 100") !=
-1 or msg.find("Hair Spa, Price: 150") != -1 or msg.find("Head Massage, Price: 150") !=
-1 or msg.find("Hair Styling, Price: 150") != -1 or msg.find("Hair Coloring, Price: 20
0") != -1 or msg.find("Shave + Haircut, Price: 150") != -1 or msg.find("Head Massage +
 Shave, Price: 200") != -1  ):
            user_catalog_choice = msg
            bot.send_message(message.chat.id, "Now lets choose the timing (type 'Choose
time')")


        else:
            bot.send_message(message.chat.id, reply)

        print("\nuser name : " + name)
        print("user message : " + msg)
        print("robo message : " + reply)
```

In [ ]:

```
while True:
    try:
        bot.polling()
    except Exception:
        time.sleep(15)
```

```
user name : Sahil
user message : /start
robo message : Hello! This is a chatbot and my name is Lucas. I will guide
you through the booking process.
As you know due to this current pandemic situation people are avoiding sal
ons to dont get in contact with the virus.

So here i offer you a online booking system , so there is no need of any h
assle or waiting in queue.

At the given booking time only one person will be handeled and we assure y
ou totally sanitized and good hygenic environment. /help.

user name : Sahil
user message : /help
robo message : How may i help you ?

user name : Sahil
user message : Hello
robo message : Hi dear, Wellcome to 'BigMan Saloon'. How can i help you.

user name : Sahil
user message : I would like to book a slot
robo message : Yes seats are available(type 'Menu')

user name : Sahil
user message : Menu
robo message : Catalog

user name : Sahil
user message : Menu
robo message : Yes i will show you the catalog

user name : Sahil
user message : Shave + Haircut, Price: 150
robo message : Ofcourse there are seats for you(type 'Menu')

user name : Sahil
user message : Time
robo message : Timing

user name : Sahil
user message : Time
robo message : We have provided flexible timeslots.

user name : Sahil
user message : 10 AM
robo message : Ofcourse there are seats for you(type 'Menu')

user name : Sahil
user message : Chose date
robo message : Date

user name : Sahil
user message : Chose date
robo message : Yes Yes it is available (type 'Menu')

user name : Sahil
user message : 25
robo message : Yes Yes it is available (type 'Menu')
```

```
user name : Sahil
user message : Confirm
robo message : Please confirm.

user name : Sahil
user message : YES
robo message : Please while ticket is being generated.

user name : Sahil
user message : No thank you
robo message : Payment needs to be done at the solon. Online payment at th
e salon can be done.
Happy to help ☺
```

In [ ]:

In [ ]: