Final Project - Disease Classifier

In [1]:
```
# Importing all the Necessary Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import pyttsx3

from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split

from sklearn import metrics
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.preprocessing import StandardScaler
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC

np.random.seed(0)

import torch
import csv
import torch.optim as optim
import torch.nn as nn
from collections import OrderedDict
from sklearn.metrics import mean_squared_error
```

In [2]:
```
# Reading in the "dataset.csv" File.
ds = pd.read_csv('./dataset.csv')

# Replacing all the Nan values with a 0 as a place holder
ds.replace(np.nan, 0, inplace=True)

#Replacing all the Symptoms(Strings) into Integers,
#with each of the symptom having a unique integer  ranging from (1 to 134)
for col in ds.columns:
    ds.replace(('itching',' skin_rash',' nodal_skin_eruptions',' continuous_sneezing',
                ' joint_pain',' stomach_pain',' acidity',' ulcers_on_tongue',' muscle_
                ' burning_micturition',' spotting_ urination',' fatigue',' weight_gair
                ' cold_hands_and_feets',' mood_swings',' weight_loss',' restlessness',
                ' irregular_sugar_level',' cough',' high_fever',' sunken_eyes',' breat
                ' dehydration',' indigestion',' headache',' yellowish_skin',' dark_uri
                ' pain_behind_the_eyes',' back_pain',' constipation',' abdominal_pain'
                ' yellow_urine',' yellowing_of_eyes',' acute_liver_failure',' fluid_ov
                ' swelled_lymph_nodes',' malaise',' blurred_and_distorted_vision',' ph
                ' redness_of_eyes',' sinus_pressure',' runny_nose',' congestion',' che
                ' fast_heart_rate',' pain_during_bowel_movements',' pain_in_anal_regio
                ' irritation_in_anus',' neck_pain',' dizziness',' cramps',' bruising',
                ' swollen_blood_vessels',' puffy_face_and_eyes',' enlarged_thyroid','
                ' swollen_extremeties',' excessive_hunger',' extra_marital_contacts','
                ' slurred_speech',' knee_pain',' hip_joint_pain',' muscle_weakness','
                ' movement_stiffness',' spinning_movements',' loss_of_balance',' unste
                ' weakness_of_one_body_side',' loss_of_smell',' badder_discomfort',' f
                ' continuous_feel_of_urine',' passage_of_gases',' internal_itching','
                ' irritability',' muscle_pain',' altered_sensorium',' red_spots_over_b
```

```
                ' abnormal_menstruation',' dischromic _patches',' watering_from_eyes',
                ' family_history',' mucoid_sputum',' rusty_sputum',' lack_of_concentra
                ' receiving_blood_transfusion',' receiving_unsterile_injections',' com
                ' distention_of_abdomen',' history_of_alcohol_consumption',' fluid_ove
                ' prominent_veins_on_calf',' palpitations',' painful_walking',' pus_fi
                ' scurring',' skin_peeling',' silver_like_dusting',' small_dents_in_na
                ' blister',' red_sore_around_nose',' yellow_crust_ooze',' prognosis','
               (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
                32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,5
                66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,8
                100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,11
                inplace=True)
```

In [3]:
```
##Replacing all the Diseases(Strings) into Integers,
#with each of the disease having a unique integer  ranging from (1 to 41)
ds.Disease.replace(('Fungal infection','Allergy','GERD','Chronic cholestasis','Drug Re
                    'AIDS','Diabetes ','Gastroenteritis','Bronchial Asthma','Hypertens
                    'Cervical spondylosis','Paralysis (brain hemorrhage)','Jaundice','
                    'Typhoid','hepatitis A','Hepatitis B','Hepatitis C','Hepatitis D',
                    'Alcoholic hepatitis','Tuberculosis','Common Cold','Pneumonia','Di
                    'Heart attack','Varicose veins','Hypothyroidism','Hyperthyroidism'
                    'Osteoarthristis','Arthritis','(vertigo) Paroymsal  Positional Ver
                    'Urinary tract infection','Psoriasis','Impetigo'),
                   (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,
                    21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41), i
```

In [4]:
```
ds
```

Out[4]:

	Disease	Symptom_1	Symptom_2	Symptom_3	Symptom_4	Symptom_5	Symptom_6	Sympton
0	1	1	2	3	103	0	0	
1	1	2	3	103	0	0	0	
2	1	1	3	103	0	0	0	
3	1	1	2	103	0	0	0	
4	1	1	2	3	0	0	0	
...	...	...	...	...	...	...	...	
4915	37	12	32	35	85	86	87	
4916	38	2	123	124	125	0	0	
4917	39	13	134	91	92	0	0	
4918	40	2	7	126	127	128	129	
4919	41	2	26	130	131	132	0	

4920 rows × 18 columns

In [5]:
```
# Splitting the Data into X - Inputs (Symptoms) and Y - Output (Disease)
N = ds.shape[1]
X = ds.values[:,1 :N]
Y = ds.iloc[:, 0].values
```

In [6]:
```
X
```

Out[6]:
```
array([[  1,   2,   3, ...,   0,   0,   0],
       [  2,   3, 103, ...,   0,   0,   0],
       [  1,   3, 103, ...,   0,   0,   0],
       ...,
       [ 13, 134,  91, ...,   0,   0,   0],
       [  2,   7, 126, ...,   0,   0,   0],
       [  2,  26, 130, ...,   0,   0,   0]], dtype=int64)
```

In [7]:
```
Y
```

Out[7]:
```
array([ 1,  1,  1, ..., 39, 40, 41], dtype=int64)
```

In [8]:
```
X_train, X_val, Y_train, Y_val = train_test_split(X, Y, train_size = 0.8, test_size =
```

In [9]:
```
# Naive Bayes Classifier
classifier = GaussianNB()
classifier.fit(X_train, Y_train)

Y_pred = classifier.predict(X_val)

# Evaluation of the Model for Accuracy, Precision, and Recall
accuracy = metrics.accuracy_score(Y_val, Y_pred)
precision = metrics.precision_score(Y_val, Y_pred, average = 'micro')
recall = metrics.recall_score(Y_val, Y_pred, average = 'micro')

print('Accuracy:', accuracy)
print('Precision:', precision)
print('Recall:', recall)
```

```
Accuracy: 0.9796747967479674
Precision: 0.9796747967479674
Recall: 0.9796747967479674
```

In [10]:
```
severityDictionary=dict()
description_list = dict()
precautionDictionary=dict()
```

In [ ]:
```
```

In [11]:
```
def getDescription():
    global description_list
    with open('symptom_Description.csv') as csv_file:
        csv_reader = csv.reader(csv_file, delimiter=',')
        line_count = 0
        for row in csv_reader:
            _description={row[0]:row[1]}
            description_list.update(_description)
```

In [ ]:
```
```

In [ ]:

In [12]:
```
def sec_predict(symptoms_exp):
    X = df.iloc[:, :-1]
    y = df['prognosis']
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_st
    rf_clf = DecisionTreeClassifier()
    rf_clf.fit(X_train, y_train)

    symptoms_dict = {symptom: index for index, symptom in enumerate(X)}
    input_vector = np.zeros(len(symptoms_dict))
    for item in symptoms_exp:
      input_vector[[symptoms_dict[item]]] = 1

    return rf_clf.predict([input_vector])
```

In [13]:
```
def getInfo():
    ChatNAME=('Norm')
    print("-----------------------------------------------Disease Classifier ChatBot
    print(ChatNAME+": Welcome to the Niner Disease Classifier.")
    print("\n"+ChatNAME+": What is your Name? \t\t\t\t\t\t\t\t",end="-> Patient: ")
    name=input("")
    print(ChatNAME+": "+"Hello, ",name+".")
```

In [14]:
```
def getSymp():
    ChatNAME=('Norm')
    # Creating an empty list
    patient_symptoms = []

    # number of elements as input
    print("\n"+ChatNAME+": "+"How many Symmptoms do you have? \t\t\t\t\t\t\t",end="->
    n = int(input())
    print("\n\t\t\t\t    -------Enter symptoms one at a time below.-------")


    # iterating till the range
    for i in range(0, n):
        symptoms = input()
        new_symptoms= symptoms.replace(" ","_")
        new_symptoms_1=new_symptoms.lower()

        patient_symptoms.append(new_symptoms_1) # adding the elements

    #print(patient_symptoms)
    #print(type(patient_symptoms))

    new_patient_symptoms=' '.join(patient_symptoms)
    #print(new_patient_symptoms)

    new_patient_symptoms=(new_patient_symptoms.replace('itching', '1').replace('skin_r
                          .replace('nodal_skin_eruptions', '3').replace('continuous_sr
                          .replace('chills', '6').replace('joint_pain', '7').replace('
                          .replace('ulcers_on_tongue', '10').replace('muscle_wasting',
                          .replace('burning_micturition', '13').replace('spotting_urir
                          .replace('weight_gain', '16').replace('anxiety', '17').repla
                          .replace('mood_swings', '19').replace('weight_loss', '20').r
                          .replace('lethargy', '22').replace('patches_in_throat', '23'
                          .replace('cough', '25').replace('high_fever', '26').replace(
```

```
                        .replace('breathlessness', '28').replace('sweating', '29').r
                        .replace('indigestion', '31').replace('headache', '32').repl
                        .replace('dark_urine', '34').replace('nausea', '35').replace
                        .replace('pain_behind_the_eyes', '37').replace('back_pain',
                        .replace('abdominal_pain', '40').replace('diarrhoea', '41').
                        .replace('yellow_urine', '43').replace('yellowing_of_eyes',
                        .replace('fluid_overload', '46').replace('swelling_of_stomac
                        .replace('swelled_lymph_nodes', '48').replace('malaise', '49
                        .replace('blurred_and_distorted_vision', '50').replace('phle
                        .replace('throat_irritation', '52').replace('redness_of_eyes
                        .replace('runny_nose', '55').replace('congestion', '56').rep
                        .replace('weakness_in_limbs', '58').replace('fast_heart_rate
                        .replace('pain_during_bowel_movements', '60').replace('pain_
                        .replace('bloody_stool', '62').replace('irritation_in_anus',
                        .replace('dizziness', '65').replace('cramps', '66').replace(
                        .replace('swollen_legs', '69').replace('swollen_blood_vessel
                        .replace('puffy_face_and_eyes', '71').replace('enlarged_thyr
                        .replace('swollen_extremeties', '74').replace('excessive_hun
                        .replace('extra_marital_contacts', '76').replace('drying_and
                        .replace('slurred_speech', '78').replace('knee_pain', '79').
                        .replace('muscle_weakness', '81').replace('stiff_neck', '82'
                        .replace('movement_stiffness', '84').replace('spinning_moven
                        .replace('loss_of_balance', '86').replace('unsteadiness', '8
                        .replace('weakness_of_one_body_side', '88').replace('loss_of
                        .replace('badder_discomfort', '90').replace('foul_smell_of u
                        .replace('continuous_feel_of_urine', '92').replace('passage_
                        .replace('internal_itching', '94').replace('toxic_look_(typh
                        .replace('irritability', '97').replace('muscle_pain', '98').
                        .replace('red_spots_over_body', '100').replace('belly_pain',
                        .replace('abnormal_menstruation', '102').replace('dischromic
                        .replace('watering_from_eyes', '104').replace('increased_app
                        .replace('family_history', '107').replace('mucoid_sputum', '
                        .replace('lack_of_concentration', '110').replace('visual_dis
                        .replace('receiving_blood_transfusion', '112').replace('rece
                        .replace('coma', '114').replace('stomach_bleeding', '115').r
                        .replace('history_of_alcohol_consumption', '117').replace('h
                        .replace('blood_in_sputum', '119').replace('prominent_veins_
                        .replace('palpitations', '121').replace('painful_walking', '
                        .replace('blackheads', '124').replace('scurring', '125').rep
                        .replace('silver_like_dusting', '127').replace('small_dents_
                        .replace('inflammatory_nails', '129').replace('blister', '13
                        .replace('yellow_crust_ooze', '132').replace('prognosis', '1


    def Convert(string):
        li = list(string.split(" "))
        return li
    new = Convert(new_patient_symptoms)
    new.sort()

    new=' '.join(new)
    #print(new)
    #print(type(new))
    final = new.split()

    new_Y = np.array(final)
    full = len(new_Y)
    empty = 17 - full

    in_symptoms = np.zeros(17)
```

```
    for i in range(full):
        in_symptoms[i] = new_Y[i]

    in_symptoms = in_symptoms.reshape(1, -1)
    #print(new_Y)
    #print(type(new_Y))
    pred_diag = classifier.predict(in_symptoms)
    pred_diag_1 = pred_diag.tolist()
    #print(type(pred_diag_1))
    #print(pred_diag_1)
    pred_diag_1 =' '.join(str(e) for e in pred_diag_1)
    dis  = (int(pred_diag_1))
    #print(dis)
    #print(type(dis))
    #print(pred_diag_1)
    #print(type(pred_diag_1))
    if pred_diag_1 == '1':
        pred_diag_1=pred_diag_1.replace('1','Fungal Infection')
    elif pred_diag_1 == '2':
        pred_diag_1 =pred_diag_1.replace('2','Allergy')
    elif pred_diag_1 == '3':
        pred_diag_1 =pred_diag_1.replace('3', 'GERD')
    elif pred_diag_1 == '4':
        pred_diag_1 =pred_diag_1.replace('4', 'Chronic cholestasis')
    elif pred_diag_1 == '5':
        pred_diag_1 =pred_diag_1.replace('5', 'Drug Reaction')
    elif pred_diag_1 == '6':
        pred_diag_1 =pred_diag_1.replace('6', 'Peptic ulcer diseae')
    elif pred_diag_1 == '7':
        pred_diag_1 =pred_diag_1.replace('7', 'AIDS')
    elif pred_diag_1 == '8':
        pred_diag_1 =pred_diag_1.replace('8', 'Diabetes')
    elif pred_diag_1 == '9':
        pred_diag_1 =pred_diag_1.replace('9', 'Gastroenteritis')
    elif pred_diag_1 == '10':
        pred_diag_1 =pred_diag_1.replace('10', 'Bronchial Asthma')
    elif pred_diag_1 == '11':
        pred_diag_1 =pred_diag_1.replace('11', 'Hypertension')
    elif pred_diag_1 == '12':
        pred_diag_1 =pred_diag_1_1.replace('12', 'Migraine')
    elif pred_diag_1 == '13':
        pred_diag_1 =pred_diag_1.replace('13', 'Cervical spondylosis')
    elif pred_diag_1 == '14':
        pred_diag_1 =pred_diag_1.replace('14', 'Paralysis (brain hemorrhage)')
    elif pred_diag_1 == '15':
        pred_diag_1 =pred_diag_1.replace('15', 'Jaundice')
    elif pred_diag_1 == '16':
        pred_diag_1 =pred_diag_1.replace('16', 'Malaria')
    elif pred_diag_1 == '17':
        pred_diag_1 =pred_diag_1.replace('17', 'Chicken pox')
    elif pred_diag_1 == '18':
        pred_diag_1 =pred_diag_1.replace('18', 'Dengue')
    elif pred_diag_1 == '19':
        pred_diag_1 =pred_diag_1.replace('19', 'Typhoid')
    elif pred_diag_1 == '20':
        pred_diag_1 =pred_diag_1.replace('20', 'hepatitis A')
    elif pred_diag_1 == '21':
        pred_diag_1 =pred_diag_1.replace('21', 'Hepatitis B')
    elif pred_diag_1 == '22':
```

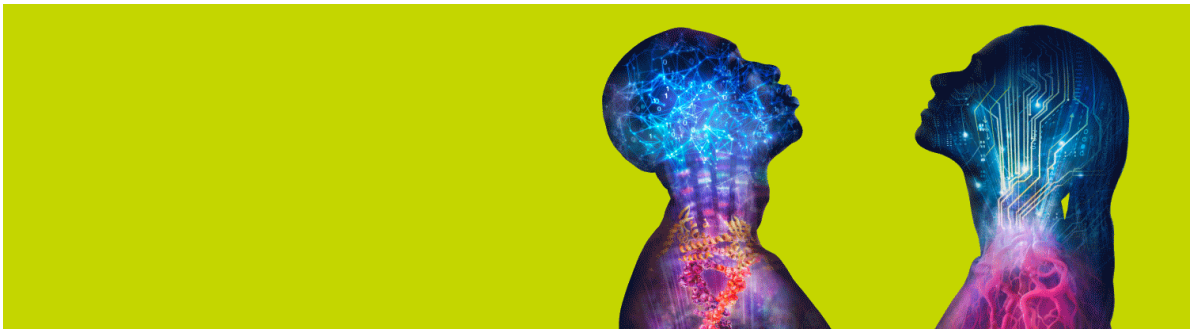
```
        pred_diag_1 =pred_diag_1.replace('22', 'Hepatitis C')
    elif pred_diag_1 == '23':
        pred_diag_1 =pred_diag_1.replace('23', 'Hepatitis D')
    elif pred_diag_1 == '24':
        pred_diag_1 =pred_diag_1.replace('24', 'Hepatitis E')
    elif pred_diag_1 == '25':
        pred_diag_1 =pred_diag_1.replace('25', 'Alcoholic hepatitis')
    elif pred_diag_1 == '26':
        pred_diag_1 =pred_diag_1.replace('26', 'Tuberculosis')
    elif pred_diag_1 == '27':
        pred_diag_1 =pred_diag_1.replace('27', 'Common Cold')
    elif pred_diag_1 == '28':
        pred_diag_1 =pred_diag_1.replace('28', 'Pneumonia')
    elif pred_diag_1 == '29':
        pred_diag_1 =pred_diag_1.replace('29', 'Dimorphic hemmorhoids(piles)')
    elif pred_diag_1 == '30':
        pred_diag_1 =pred_diag_1.replace('30', 'Heart attack')
    elif pred_diag_1 == '31':
        pred_diag_1 =pred_diag_1.replace('31', 'Varicose veins')
    elif pred_diag_1 == '32':
        pred_diag_1 =pred_diag_1.replace('32', 'Hypothyroidism')
    elif pred_diag_1 == '33':
        pred_diag_1 =pred_diag_1.replace('33', 'Hyperthyroidism')
    elif pred_diag_1 == '34':
        pred_diag_1 =pred_diag_1.replace('34', 'Hypoglycemia')
    elif pred_diag_1 == '35':
        pred_diag_1 =pred_diag_1.replace('35', 'Osteoarthristis')
    elif pred_diag_1 == '36':
        pred_diag_1 =pred_diag_1.replace('36', 'Arthritis')
    elif pred_diag_1 == '37':
        pred_diag_1 =pred_diag_1.replace('37', '(vertigo) Paroymsal  Positional Vertig
    elif pred_diag_1 == '38':
        pred_diag_1 =pred_diag_1.replace('38', 'Acne')
    elif pred_diag_1 == '39':
        pred_diag_1 =pred_diag_1.replace('39', 'Urinary tract infection')
    elif pred_diag_1 == '40':
        pred_diag_1 =pred_diag_1.replace('40', 'Psoriasis')
    elif pred_diag_1 == '41':
        pred_diag_1 =pred_diag_1.replace('41', 'Impetigo')


    import itertools
    import threading
    import time
    import sys
    def process():
        print("\n\t\t\t\t    ------------------------------------------------")
        print("\t\t\t\t\t\t\t Processing:")
        #animation = ["10%", "20%", "30%", "40%", "50%", "60%", "70%", "80%", "90%", "
        animation = ["■□□□□□□□□□ 10%","■■□□□□□□□□ 20%", "■■■□□□□□□□ 30%", "■■■■□□□□□□
                     "■■■■■■■□□□□ 60%", "■■■■■■■■□□□ 70%","■■■■■■■■■□□ 80%", "■■■■■■■■■■□

        for i in range(len(animation)):
            time.sleep(0.5)
            sys.stdout.write("\r" + "\t\t\t\t\t\t\t\t"+animation[i % len(animation)])
            sys.stdout.flush()
        print("\n\t\t\t\t\t\t\t     Processing Complete:")
        print("\t\t\t\t    ------------------------------------------------")
        print("\n")
```

```
    process()
    import math
    print(ChatNAME+": Your predeicted disease is: {} with {}% accuracy.".format(pred_d

    def dis_desc(dis):
        # Read in data file.
        desc = pd.read_csv('./symptom_Description.csv')

        # Convert descriptions to an array.
        D = desc.iloc[:, 1].values

        # Print description at index of predicted disease.
        print("\n"+ChatNAME+": Disease Description:")
        print("\t"+D[dis-1])

    def next_steps(dis):
        # Read in data file.
        prec = pd.read_csv('./symptom_precaution.csv')

        # Convert descriptions to an array.
        P = prec.iloc[:,1:5].values

        # Print the suggested next steps at index of predicted disease.
        print("\n"+ChatNAME+": Suggested Precautions:")
        for i in range(0,4):
            out = P[dis-1,i]
            print("\t{}. {}".format(i+1, out))

    #print(dis)
    print("\n"+ChatNAME+": Would you like a description of your disease?\t\t\t\t\t",er
    ans_1= input()
    ans_1= ans_1.lower()
    if ans_1 == 'yes':
        dis_desc(dis)
    elif ans_1 == 'no':
        print("\n"+ChatNAME+": Would you like suggested precautions for your disease?\
        ans_2=input()
        ans_2=ans_2.lower()
        if ans_2 == 'yes':
            next_steps(dis)
        if ans_2 == 'no':
            return



    print("\n"+ChatNAME+": Would you like suggested precautions for your disease?\t\t\
    ans_2=input()
    ans_2=ans_2.lower()
    if ans_2 == 'yes':
        next_steps(dis)
    if ans_2 == 'no':
        return

import os
```

In [15]:

```
import webbrowser
def end():
    ChatNAME=('Norm')
    print("\n"+ChatNAME+": Thank you for using the Disease Classifier ChatBot.")
```

```
    print(ChatNAME+": Would you like to schedule a appointment at the Niner Health Cer
    ans = input()
    ans = ans.lower()
    if ans == 'yes':
          webbrowser.open("https://uncc.medicatconnect.com/Shibboleth.sso/Login?entity
    elif ans == 'no':
        print("\n"+ChatNAME+": Have a good day, goodbye.")
```



In [16]:
```
getInfo(),getSymp(),end()
print("--------------------------------------------------------------------------------
```

```
--------------------------------------------------Disease Classifier ChatBot----------
----------------------------------------
Norm: Welcome to the Niner Disease Classifier.

Norm: What is your Name?
-> Patient: Noah
Norm: Hello,  Noah.

Norm: How many Symmptoms do you have?
-> Patient: 2

                                       -------Enter symptoms one at a time below.-------
chills
cough

                                       --------------------------------------------------
                                                        Processing:
                                                      ■■■■■■■■■■ 100%
                                                   Processing Complete:
                                       --------------------------------------------------


Norm: Your predeicted disease is: AIDS with 97.97% accuracy.

Norm: Would you like a description of your disease?
-> Patient: yes

Norm: Disease Description:
        Acquired immunodeficiency syndrome (AIDS) is a chronic, potentially life-thre
atening condition caused by the human immunodeficiency virus (HIV). By damaging your
immune system, HIV interferes with your body's ability to fight infection and diseas
e.

Norm: Would you like suggested precautions for your disease?
-> Patient: yes

Norm: Suggested Precautions:
        1. avoid open cuts
        2. wear ppe if possible
        3. consult doctor
        4. follow up

Norm: Thank you for using the Disease Classifier ChatBot.
Norm: Would you like to schedule a appointment at the Niner Health Center?
-> Patient: yes
--------------------------------------------------------------------------------------
----------------------------------------
```

In [ ]:

In [ ]: