Name- Dev Chauhan

University rollno.- 2013648

Class rollno. - 29

Disease Prediction using Machine Learning

In [1]:

```
#Importing Libraries
from sklearn.preprocessing import StandardScaler
from tkinter import *
import numpy as np
import pandas as pd
```

In [2]:

```
#List of the symptoms is listed here in list l1.

l1=['back_pain','constipation','abdominal_pain','diarrhoea','mild_fever','yellow_urine',
    'yellowing_of_eyes','acute_liver_failure','fluid_overload','swelling_of_stomach',
    'swelled_lymph_nodes','malaise','blurred_and_distorted_vision','phlegm','throat_irritat
    'redness_of_eyes','sinus_pressure','runny_nose','congestion','chest_pain','weakness_in_
    'fast_heart_rate','pain_during_bowel_movements','pain_in_anal_region','bloody_stool',
    'irritation_in_anus','neck_pain','dizziness','cramps','bruising','obesity','swollen_leg
    'swollen_blood_vessels','puffy_face_and_eyes','enlarged_thyroid','brittle_nails',
    'swollen_extremeties','excessive_hunger','extra_marital_contacts','drying_and_tingling_
    'slurred_speech','knee_pain','hip_joint_pain','muscle_weakness','stiff_neck','swelling_
    'movement_stiffness','spinning_movements','loss_of_balance','unsteadiness',
    'weakness_of_one_body_side','loss_of_smell','bladder_discomfort','foul_smell_of urine',
    'continuous_feel_of_urine','passage_of_gases','internal_itching','toxic_look_(typhos)',
    'depression','irritability','muscle_pain','altered_sensorium','red_spots_over_body','be
    'abnormal_menstruation','dischromic _patches','watering_from_eyes','increased_appetite'
    'rusty_sputum','lack_of_concentration','visual_disturbances','receiving_blood_transfusi
    'receiving_unsterile_injections','coma','stomach_bleeding','distention_of_abdomen',
    'history_of_alcohol_consumption','fluid_overload','blood_in_sputum','prominent_veins_on
    'palpitations','painful_walking','pus_filled_pimples','blackheads','scurring','skin_pee
    'silver_like_dusting','small_dents_in_nails','inflammatory_nails','blister','red_sore_a
    'yellow_crust_ooze']
```

In [3]:

```
#List of Diseases is listed in list disease.

disease=['Fungal infection', 'Allergy', 'GERD', 'Chronic cholestasis',
       'Drug Reaction', 'Peptic ulcer diseae', 'AIDS', 'Diabetes ',
       'Gastroenteritis', 'Bronchial Asthma', 'Hypertension ', 'Migraine',
       'Cervical spondylosis', 'Paralysis (brain hemorrhage)', 'Jaundice',
       'Malaria', 'Chicken pox', 'Dengue', 'Typhoid', 'hepatitis A',
       'Hepatitis B', 'Hepatitis C', 'Hepatitis D', 'Hepatitis E',
       'Alcoholic hepatitis', 'Tuberculosis', 'Common Cold', 'Pneumonia',
       'Dimorphic hemmorhoids(piles)', 'Heart attack', 'Varicose veins',
       'Hypothyroidism', 'Hyperthyroidism', 'Hypoglycemia',
       'Osteoarthristis', 'Arthritis',
       '(vertigo) Paroymsal  Positional Vertigo', 'Acne',
       'Urinary tract infection', 'Psoriasis', 'Impetigo']
```

In [4]:

```
l2=[]
for i in range(0,len(l1)):
    l2.append(0)
print(l2)
```

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

In [5]:

```
#Reading the training .csv file
df=pd.read_csv("training.csv")
DF= pd.read_csv('training.csv', index_col='prognosis')
#Replace the values.

df.replace({'prognosis':{'Fungal infection':0,'Allergy':1,'GERD':2,'Chronic cholestasis':3,
    'Peptic ulcer diseae':5,'AIDS':6,'Diabetes ':7,'Gastroenteritis':8,'Bronchial Asthma':9
    'Migraine':11,'Cervical spondylosis':12,
    'Paralysis (brain hemorrhage)':13,'Jaundice':14,'Malaria':15,'Chicken pox':16,'Dengue':
    'Hepatitis B':20,'Hepatitis C':21,'Hepatitis D':22,'Hepatitis E':23,'Alcoholic hepatiti
    'Common Cold':26,'Pneumonia':27,'Dimorphic hemmorhoids(piles)':28,'Heart attack':29,'Va
    'Hyperthyroidism':32,'Hypoglycemia':33,'Osteoarthristis':34,'Arthritis':35,
    '(vertigo) Paroymsal  Positional Vertigo':36,'Acne':37,'Urinary tract infection':38,'Ps
    'Impetigo':40}},inplace=True)

DF.head(20)
```

Out[5]:

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joir
prognosis							
Fungal infection	1	1	1	0	0	0	
Fungal infection	0	1	1	0	0	0	
Fungal infection	1	0	1	0	0	0	
Fungal infection	1	1	0	0	0	0	
Fungal infection	1	1	1	0	0	0	
Fungal infection	0	1	1	0	0	0	
Fungal infection	1	0	1	0	0	0	
Fungal infection	1	1	0	0	0	0	
Fungal infection	1	1	1	0	0	0	
Fungal infection	1	1	1	0	0	0	
Allergy	0	0	0	1	1	1	
Allergy	0	0	0	0	1	1	
Allergy	0	0	0	1	0	1	
Allergy	0	0	0	1	1	0	
Allergy	0	0	0	1	1	1	
Allergy	0	0	0	0	1	1	
Allergy	0	0	0	1	0	1	
Allergy	0	0	0	1	1	0	

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joir
prognosis							
Allergy	0	0	0	1	1	1	
Allergy	0	0	0	1	1	1	

20 rows × 133 columns

In [6]:

```
X= df[l1]
y = df[["prognosis"]]
np.ravel(y)
print(X)
```

```
      back_pain  constipation  abdominal_pain  diarrhoea  mild_fever  \
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
...         ...           ...             ...        ...         ...
4915          0             0               0          0           0
4916          0             0               0          0           0
4917          0             0               0          0           0
4918          0             0               0          0           0
4919          0             0               0          0           0

      yellow_urine  yellowing_of_eyes  acute_liver_failure  fluid_overload
\
0                0                  0                    0               0
1                0                  0                    0               0
2                0                  0                    0               0
3                0                  0                    0               0
4                0                  0                    0               0
...            ...                ...                  ...             ...
4915             0                  0                    0               0
4916             0                  0                    0               0
4917             0                  0                    0               0
4918             0                  0                    0               0
4919             0                  0                    0               0

      swelling_of_stomach  ...  pus_filled_pimples  blackheads  scurring  \
0                       0  ...                   0           0         0
1                       0  ...                   0           0         0
2                       0  ...                   0           0         0
3                       0  ...                   0           0         0
4                       0  ...                   0           0         0
...                   ...  ...                 ...         ...       ...
4915                    0  ...                   0           0         0
4916                    0  ...                   1           1         1
4917                    0  ...                   0           0         0
4918                    0  ...                   0           0         0
4919                    0  ...                   0           0         0

      skin_peeling  silver_like_dusting  small_dents_in_nails  \
0                0                    0                     0
1                0                    0                     0
2                0                    0                     0
3                0                    0                     0
4                0                    0                     0
...            ...                  ...                   ...
4915             0                    0                     0
4916             0                    0                     0
4917             0                    0                     0
4918             1                    1                     1
4919             0                    0                     0

      inflammatory_nails  blister  red_sore_around_nose  yellow_crust_ooze
```

```
0                      0        0                     0                  0
1                      0        0                     0                  0
2                      0        0                     0                  0
3                      0        0                     0                  0
4                      0        0                     0                  0
...                  ...      ...                   ...                ...
4915                   0        0                     0                  0
4916                   0        0                     0                  0
4917                   0        0                     0                  0
4918                   1        0                     0                  0
4919                   0        1                     1                  1

[4920 rows x 95 columns]
```

In [7]:

```
print(y)
```

```
      prognosis
0             0
1             0
2             0
3             0
4             0
...         ...
4915         36
4916         37
4917         38
4918         39
4919         40

[4920 rows x 1 columns]
```

In [8]:

```
#Reading the  testing.csv file
tr=pd.read_csv("testing.csv")

#Using inbuilt function replace in pandas for replacing the values

tr.replace({'prognosis':{'Fungal infection':0,'Allergy':1,'GERD':2,'Chronic cholestasis':3,
    'Peptic ulcer diseae':5,'AIDS':6,'Diabetes ':7,'Gastroenteritis':8,'Bronchial Asthma':9
    'Migraine':11,'Cervical spondylosis':12,
    'Paralysis (brain hemorrhage)':13,'Jaundice':14,'Malaria':15,'Chicken pox':16,'Dengue':
    'Hepatitis B':20,'Hepatitis C':21,'Hepatitis D':22,'Hepatitis E':23,'Alcoholic hepatiti
    'Common Cold':26,'Pneumonia':27,'Dimorphic hemmorhoids(piles)':28,'Heart attack':29,'Va
    'Hyperthyroidism':32,'Hypoglycemia':33,'Osteoarthristis':34,'Arthritis':35,
    '(vertigo) Paroymsal  Positional Vertigo':36,'Acne':37,'Urinary tract infection':38,'Ps
    'Impetigo':40}},inplace=True)
tr.head()
```

Out[8]:

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain
0	1	1	1	0	0	0	0
1	0	0	0	1	1	1	0
2	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0
4	1	1	0	0	0	0	0

5 rows × 133 columns

In [9]:

```
X_test= tr[l1]
y_test = tr[["prognosis"]]
np.ravel(y_test)
print(X_test)
```

```
26              0               0                0
27              0               0                0
28              0               0                0
29              0               0                0
30              0               0                0
31              0               0                0
32              0               0                0
33              0               0                0
34              0               0                0
35              0               0                0
36              0               0                0
37              0               0                0
38              0               0                0
39              1               1                1
40              0               0                0

    inflammatory_nails  blister  red_sore_around_nose  yellow_crust_ooze
0                    0        0                     0                  0
1                    0        0                     0                  0
2                    0        0                     0                  0
```

In [10]:

```
print(y_test)
```

```
    prognosis
0           0
1           1
2           2
3           3
4           4
5           5
6           6
7           7
8           8
9           9
10         10
11         11
12         12
13         13
14         14
15         15
16         16
17         17
18         18
19         19
20         20
21         21
22         22
23         23
24         24
25         25
26         26
27         27
28         28
29         29
30         30
31         31
32         32
33         33
34         34
35         35
36         36
37         37
38         38
39         39
40         40
```

In [11]:

```
# DecisionTree

root = Tk()
pred1=StringVar()
def DecisionTree():
    if len(NameEn.get()) == 0:
        pred1.set(" ")
        comp=messagebox.askokcancel("System","Kindly Fill the Name")
        if comp:
            root.mainloop()
    elif((Symptom1.get()=="Select Here") or (Symptom2.get()=="Select Here")):
        pred1.set(" ")
        sym=messagebox.askokcancel("System","Kindly Fill atleast first two Symptoms")
        if sym:
            root.mainloop()
    else:
        from sklearn import tree

        clf3 = tree.DecisionTreeClassifier()
        clf3 = clf3.fit(X,y)

        from sklearn.metrics import classification_report,accuracy_score
        y_pred=clf3.predict(X_test)
        print("Decision Tree")
        print("Accuracy")
        print(accuracy_score(y_test, y_pred))



        psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.g

        for k in range(0,len(l1)):
            for z in psymptoms:
                if(z==l1[k]):
                    l2[k]=1

        inputtest = [l2]
        predict = clf3.predict(inputtest)
        predicted=predict[0]

        h='no'
        for a in range(0,len(disease)):
            if(predicted == a):
                h='yes'
                break


        if (h=='yes'):
            pred1.set(" ")
            pred1.set(disease[a])
        else:
            pred1.set(" ")
            pred1.set("Not Found")
```

In [12]:

```
# Randomforest

pred2=StringVar()
def randomforest():
    if len(NameEn.get()) == 0:
        pred1.set(" ")
        comp=messagebox.askokcancel("System","Kindly Fill the Name")
        if comp:
            root.mainloop()
    elif((Symptom1.get()=="Select Here") or (Symptom2.get()=="Select Here")):
        pred1.set(" ")
        sym=messagebox.askokcancel("System","Kindly Fill atleast first two Symptoms")
        if sym:
            root.mainloop()
    else:
        from sklearn.ensemble import RandomForestClassifier
        clf4 = RandomForestClassifier(n_estimators=100)
        clf4 = clf4.fit(X,np.ravel(y))


        from sklearn.metrics import classification_report,accuracy_score
        y_pred=clf4.predict(X_test)
        print("Random Forest")
        print("Accuracy")
        print(accuracy_score(y_test, y_pred))



        psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.g

        for k in range(0,len(l1)):
            for z in psymptoms:
                if(z==l1[k]):
                    l2[k]=1

        inputtest = [l2]
        predict = clf4.predict(inputtest)
        predicted=predict[0]

        h='no'
        for a in range(0,len(disease)):
            if(predicted == a):
                h='yes'
                break
        if (h=='yes'):
            pred2.set(" ")
            pred2.set(disease[a])
        else:
            pred2.set(" ")
            pred2.set("Not Found")
```

In [13]:

```
# K Nearest Neighbour

pred4=StringVar()
def KNN():
    if len(NameEn.get()) == 0:
        pred1.set(" ")
        comp=messagebox.askokcancel("System","Kindly Fill the Name")
        if comp:
            root.mainloop()
    elif((Symptom1.get()=="Select Here") or (Symptom2.get()=="Select Here")):
        pred1.set(" ")
        sym=messagebox.askokcancel("System","Kindly Fill atleast first two Symptoms")
        if sym:
            root.mainloop()
    else:
        from sklearn.neighbors import KNeighborsClassifier
        knn=KNeighborsClassifier(n_neighbors=5,metric='minkowski',p=2)
        knn=knn.fit(X,np.ravel(y))

        from sklearn.metrics import classification_report,accuracy_score
        y_pred=knn.predict(X_test)
        print("kNearest Neighbour")
        print("Accuracy")
        print(accuracy_score(y_test, y_pred))



        psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.g

        for k in range(0,len(l1)):
            for z in psymptoms:
                if(z==l1[k]):
                    l2[k]=1

        inputtest = [l2]
        predict = knn.predict(inputtest)
        predicted=predict[0]

        h='no'
        for a in range(0,len(disease)):
            if(predicted == a):
                h='yes'
                break


        if (h=='yes'):
            pred4.set(" ")
            pred4.set(disease[a])
        else:
            pred4.set(" ")
            pred4.set("Not Found")
```

In [14]:

```
#  NaiveBayes

pred3=StringVar()
def NaiveBayes():
    if len(NameEn.get()) == 0:
        pred1.set(" ")
        comp=messagebox.askokcancel("System","Kindly Fill the Name")
        if comp:
            root.mainloop()
    elif((Symptom1.get()=="Select Here") or (Symptom2.get()=="Select Here")):
        pred1.set(" ")
        sym=messagebox.askokcancel("System","Kindly Fill atleast first two Symptoms")
        if sym:
            root.mainloop()
    else:
        from sklearn.naive_bayes import GaussianNB
        gnb = GaussianNB()
        gnb=gnb.fit(X,np.ravel(y))

        from sklearn.metrics import classification_report,accuracy_score
        y_pred=gnb.predict(X_test)
        print("Naive Bayes")
        print("Accuracy")
        print(accuracy_score(y_test, y_pred))


        psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.g
        for k in range(0,len(l1)):
            for z in psymptoms:
                if(z==l1[k]):
                    l2[k]=1

        inputtest = [l2]
        predict = gnb.predict(inputtest)
        predicted=predict[0]

        h='no'
        for a in range(0,len(disease)):
            if(predicted == a):
                h='yes'
                break
        if (h=='yes'):
            pred3.set(" ")
            pred3.set(disease[a])
        else:
            pred3.set(" ")
            pred3.set("Not Found")
```

In [15]:

```
#Tk class is used to create a root window
root.configure(background='white')
root.title('Disease Predictor')
root.resizable(0,0)
```

Out[15]:

```
''
```

In [16]:

```
Symptom1 = StringVar()
Symptom1.set("Select Here")

Symptom2 = StringVar()
Symptom2.set("Select Here")

Symptom3 = StringVar()
Symptom3.set("Select Here")

Symptom4 = StringVar()
Symptom4.set("Select Here")

Symptom5 = StringVar()
Symptom5.set("Select Here")
Name = StringVar()
```

In [17]:

```
prev_win=None
def Reset():
    global prev_win

    Symptom1.set("Select Here")
    Symptom2.set("Select Here")
    Symptom3.set("Select Here")
    Symptom4.set("Select Here")
    Symptom5.set("Select Here")
    NameEn.delete(first=0,last=100)
    pred1.set(" ")
    pred2.set(" ")
    pred3.set(" ")
    pred4.set(" ")
    try:
        prev_win.destroy()
        prev_win=None
    except AttributeError:
        pass
```

In [18]:

```
from tkinter import messagebox
def Exit():
    qExit=messagebox.askyesno("System","Do you want to exit the system")

    if qExit:
        root.destroy()
        exit()
```

In [19]:

```
#Headings for the GUI written at the top of GUI
w2 = Label(root, justify=LEFT, text="Disease Predictor", fg="Red")
w2.config(font=("comic sans",30,"bold italic"))
w2.grid(row=1, column=0, columnspan=2, padx=100)
```

In [20]:

```
#Label for the name
NameLb = Label(root, text="Name of the Patient(M) ", fg="Red", bg="Ivory")
NameLb.config(font=("Times",15,"bold italic"))
NameLb.grid(row=6, column=0, pady=15, sticky=W)
```

In [21]:

```
#Creating Labels for the symtoms
S1Lb = Label(root, text="Symptom 1(M)", fg="Black", bg="Ivory")
S1Lb.config(font=("Times",15,"bold italic"))
S1Lb.grid(row=7, column=0, pady=10, sticky=W)

S2Lb = Label(root, text="Symptom 2(M)", fg="Black", bg="Ivory")
S2Lb.config(font=("Times",15,"bold italic"))
S2Lb.grid(row=8, column=0, pady=10, sticky=W)

S3Lb = Label(root, text="Symptom 3", fg="Black",bg="Ivory")
S3Lb.config(font=("Times",15,"bold italic"))
S3Lb.grid(row=9, column=0, pady=10, sticky=W)

S4Lb = Label(root, text="Symptom 4", fg="Black", bg="Ivory")
S4Lb.config(font=("Times",15,"bold italic"))
S4Lb.grid(row=10, column=0, pady=10, sticky=W)

S5Lb = Label(root, text="Symptom 5", fg="Black", bg="Ivory")
S5Lb.config(font=("Times",15,"bold italic"))
S5Lb.grid(row=11, column=0, pady=10, sticky=W)
```

In [22]:

```
#Labels for the different algorithms
lrLb = Label(root, text="DecisionTree", fg="blue", bg="white", width = 20)
lrLb.config(font=("Times",15,"bold italic"))
lrLb.grid(row=21, column=0, pady=10,sticky=W)

destreeLb = Label(root, text="RandomForest", fg="blue", bg="white", width = 20)
destreeLb.config(font=("Times",15,"bold italic"))
destreeLb.grid(row=23, column=0, pady=10, sticky=W)

ranfLb = Label(root, text="NaiveBayes", fg="blue", bg="white", width = 20)
ranfLb.config(font=("Times",15,"bold italic"))
ranfLb.grid(row=25, column=0, pady=10, sticky=W)

knnLb = Label(root, text="kNearestNeighbour", fg="blue", bg="white", width = 20)
knnLb.config(font=("Times",15,"bold italic"))
knnLb.grid(row=27, column=0, pady=10, sticky=W)
OPTIONS = sorted(l1)
```

In [23]:

```
#Taking name as input from user
NameEn = Entry(root, textvariable=Name, width = 20, bg="light yellow", bd="6",font="arial")
NameEn.grid(row=6, column=1)

#Taking Symptoms as input from the dropdown from the user
S1 = OptionMenu(root, Symptom1,*OPTIONS)
S1.grid(row=7, column=1)

S2 = OptionMenu(root, Symptom2,*OPTIONS)
S2.grid(row=8, column=1)

S3 = OptionMenu(root, Symptom3,*OPTIONS)
S3.grid(row=9, column=1)

S4 = OptionMenu(root, Symptom4,*OPTIONS)
S4.grid(row=10, column=1)

S5 = OptionMenu(root, Symptom5,*OPTIONS)
S5.grid(row=11, column=1)
```

In [24]:

```
#Buttons for predicting the disease using different algorithms
dst = Button(root, text="Prediction 1", command=DecisionTree,bg="navy blue",fg="yellow")
dst.config(font=("Times",15,"bold italic"))
dst.grid(row=15, column=0,padx=10)

rnf = Button(root, text="Prediction 2", command=randomforest,bg="navy blue",fg="yellow")
rnf.config(font=("Times",15,"bold italic"))
rnf.grid(row=15, column=1,padx=10)

lr = Button(root, text="Prediction 3", command=NaiveBayes,bg="navy blue",fg="yellow")
lr.config(font=("Times",15,"bold italic"))
lr.grid(row=17, column=0,padx=10)

kn = Button(root, text="Prediction 4", command=KNN,bg="navy blue",fg="yellow")
kn.config(font=("Times",15,"bold italic"))
kn.grid(row=17, column=1,padx=10)

rs = Button(root,text="Reset Inputs", command=Reset,bg="pink",fg="black",width=15)
rs.config(font=("Times",15,"bold italic"))
rs.grid(row=19,column=0,padx=10)

ex = Button(root,text="Exit System", command=Exit,bg="pink",fg="black",width=15)
ex.config(font=("Times",15,"bold italic"))
ex.grid(row=19,column=1,padx=10)
```

In [25]:

```
#Showing the output of different aldorithms
t1=Label(root,font=("Times",15,"bold italic"),text="Decision Tree",height=1,bg="light blue"
         ,width=40,fg="red",textvariable=pred1,relief="sunken").grid(row=21, column=1, padx

t2=Label(root,font=("Times",15,"bold italic"),text="Random Forest",height=1,bg="light blue"
         ,width=40,fg="red",textvariable=pred2,relief="sunken").grid(row=23, column=1, padx

t3=Label(root,font=("Times",15,"bold italic"),text="Naive Bayes",height=1,bg="light blue"
         ,width=40,fg="red",textvariable=pred3,relief="sunken").grid(row=25, column=1, padx

t4=Label(root,font=("Times",15,"bold italic"),text="kNearest Neighbour",height=1,bg="light
         ,width=40,fg="red",textvariable=pred4,relief="sunken").grid(row=27, column=1, padx
```

In [26]:

```
#calling this function because the application is ready to run
root.mainloop()
```

```
Decision Tree
Accuracy
0.9512195121951219
Random Forest
Accuracy
0.9512195121951219
Naive Bayes
Accuracy
0.9512195121951219
kNearest Neighbour
Accuracy
0.926829268292683
```