Disease Prediction based on Symptoms

In [1]:

```
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
from tkinter import *
import numpy as np
import pandas as pd
import os
```

In [2]:

```
l1=['back_pain','constipation','abdominal_pain','diarrhoea','mild_feve
r','yellow_urine',
    'yellowing_of_eyes','acute_liver_failure','fluid_overload','swellin
g_of_stomach',
    'swelled_lymph_nodes','malaise','blurred_and_distorted_vision','phl
egm','throat_irritation',
    'redness_of_eyes','sinus_pressure','runny_nose','congestion','chest
_pain','weakness_in_limbs',
    'fast_heart_rate','pain_during_bowel_movements','pain_in_anal_regio
n','bloody_stool',
    'irritation_in_anus','neck_pain','dizziness','cramps','bruising','o
besity','swollen_legs',
    'swollen_blood_vessels','puffy_face_and_eyes','enlarged_thyroid','b
rittle_nails',
    'swollen_extremeties','excessive_hunger','extra_marital_contacts',
'drying_and_tingling_lips',
    'slurred_speech','knee_pain','hip_joint_pain','muscle_weakness','st
iff_neck','swelling_joints',
    'movement_stiffness','spinning_movements','loss_of_balance','unstea
diness',
    'weakness_of_one_body_side','loss_of_smell','bladder_discomfort','f
oul_smell_of urine',
    'continuous_feel_of_urine','passage_of_gases','internal_itching','t
oxic_look_(typhos)',
    'depression','irritability','muscle_pain','altered_sensorium','red_
spots_over_body','belly_pain',
    'abnormal_menstruation','dischromic _patches','watering_from_eyes',
'increased_appetite','polyuria','family_history','mucoid_sputum',
    'rusty_sputum','lack_of_concentration','visual_disturbances','recei
ving_blood_transfusion',
    'receiving_unsterile_injections','coma','stomach_bleeding','distent
ion_of_abdomen',
    'history_of_alcohol_consumption','fluid_overload','blood_in_sputum'
,'prominent_veins_on_calf',
    'palpitations','painful_walking','pus_filled_pimples','blackheads',
'scurring','skin_peeling',
    'silver_like_dusting','small_dents_in_nails','inflammatory_nails',
'blister','red_sore_around_nose',
    'yellow_crust_ooze']
```

In [3]:

```
disease=['Fungal infection', 'Allergy', 'GERD', 'Chronic cholestasis',
       'Drug Reaction', 'Peptic ulcer diseae', 'AIDS', 'Diabetes ',
       'Gastroenteritis', 'Bronchial Asthma', 'Hypertension ', 'Migrain
e',
       'Cervical spondylosis', 'Paralysis (brain hemorrhage)', 'Jaundic
e',
       'Malaria', 'Chicken pox', 'Dengue', 'Typhoid', 'hepatitis A',
       'Hepatitis B', 'Hepatitis C', 'Hepatitis D', 'Hepatitis E',
       'Alcoholic hepatitis', 'Tuberculosis', 'Common Cold', 'Pneumoni
a',
       '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]:

```
df=pd.read_csv("C:/Users/Damodhar Sai Pesay/Desktop/training.csv")
DF= pd.read_csv('C:/Users/Damodhar Sai Pesay/Desktop/training.csv', ind
ex_col='prognosis')
df.replace({'prognosis':{'Fungal infection':0,'Allergy':1,'GERD':2,'Chr
onic cholestasis':3,'Drug Reaction':4,
    'Peptic ulcer diseae':5,'AIDS':6,'Diabetes ':7,'Gastroenteritis':8,
'Bronchial Asthma':9,'Hypertension ':10,
    'Migraine':11,'Cervical spondylosis':12,
    'Paralysis (brain hemorrhage)':13,'Jaundice':14,'Malaria':15,'Chick
en pox':16,'Dengue':17,'Typhoid':18,'hepatitis A':19,
    'Hepatitis B':20,'Hepatitis C':21,'Hepatitis D':22,'Hepatitis E':23
,'Alcoholic hepatitis':24,'Tuberculosis':25,
    'Common Cold':26,'Pneumonia':27,'Dimorphic hemmorhoids(piles)':28,
'Heart attack':29,'Varicose veins':30,'Hypothyroidism':31,
    'Hyperthyroidism':32,'Hypoglycemia':33,'Osteoarthristis':34,'Arthri
tis':35,
    '(vertigo) Paroymsal  Positional Vertigo':36,'Acne':37,'Urinary tra
ct infection':38,'Psoriasis':39,
    'Impetigo':40}},inplace=True)
#df.head()
DF.head()
```

Out[5]:

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_
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	

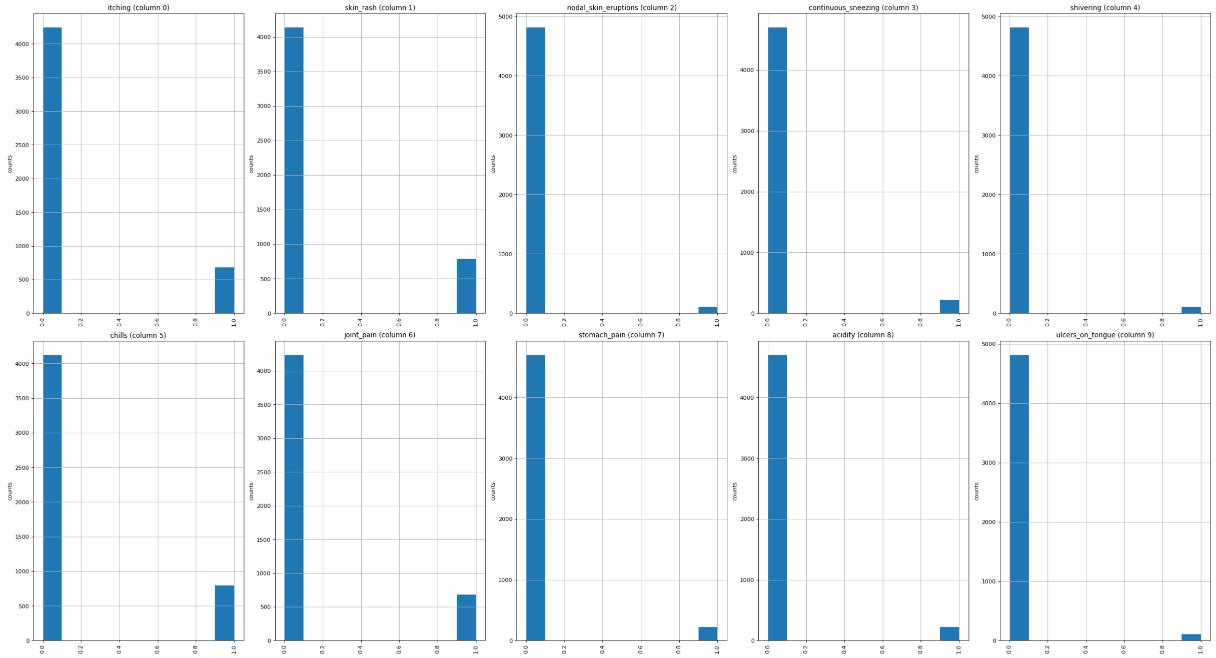
Fungal infection	1	1	1	0	0	0

5 rows × 132 columns

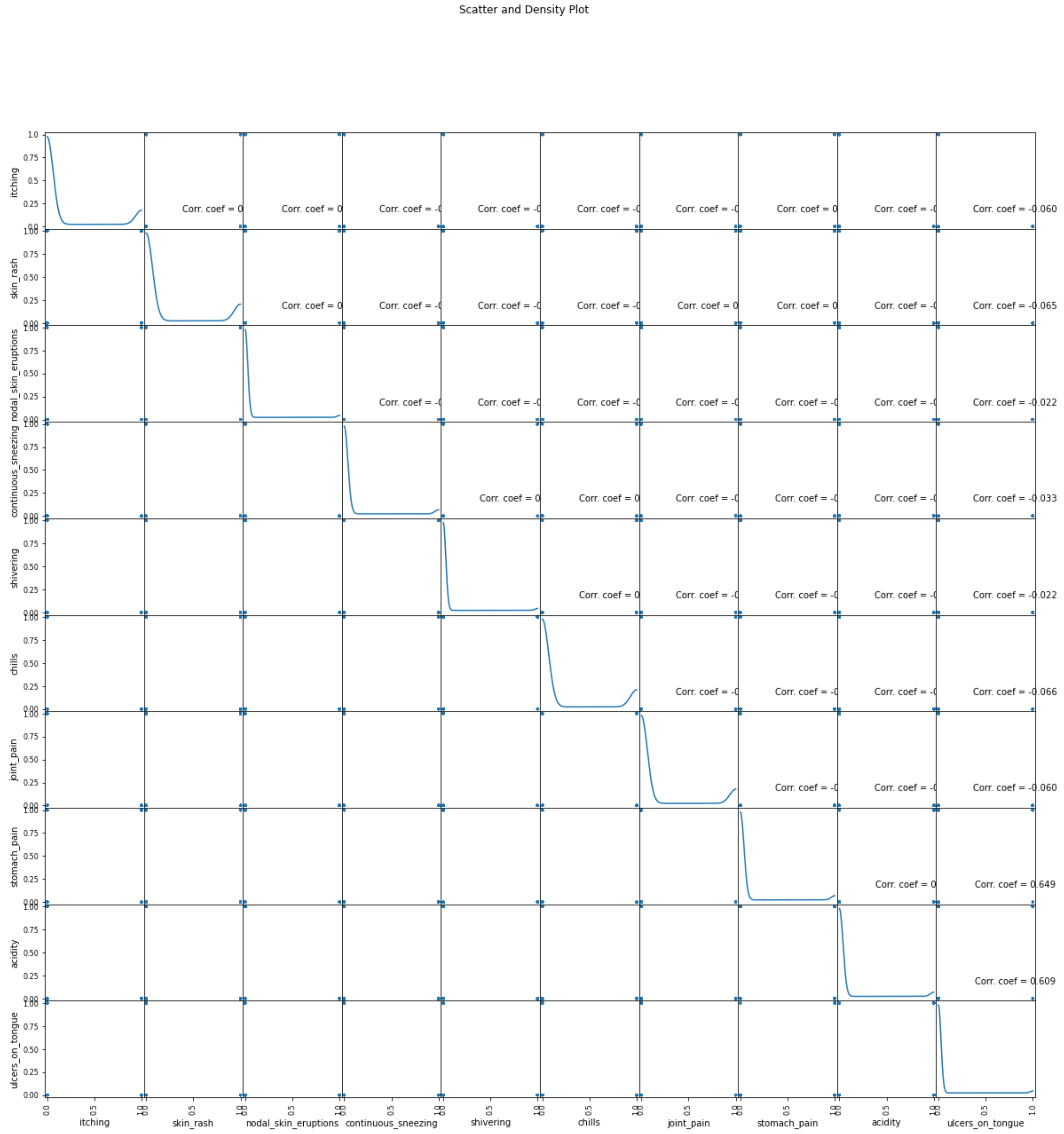
In [6]:
```
def plotPerColumnDistribution(df1, nGraphShown, nGraphPerRow):
    nunique = df1.nunique()
    df1 = df1[[col for col in df if nunique[col] > 1 and nunique[col] <
 50]]
    nRow, nCol = df1.shape
    columnNames = list(df1)
    nGraphRow = (nCol + nGraphPerRow - 1) / nGraphPerRow
    plt.figure(num = None, figsize = (6 * nGraphPerRow, 8 * nGraphRow),
 dpi = 80, facecolor = 'w', edgecolor = 'k')
    for i in range(min(nCol, nGraphShown)):
        plt.subplot(nGraphRow, nGraphPerRow, i + 1)
        columnDf = df.iloc[:, i]
        if (not np.issubdtype(type(columnDf.iloc[0]), np.number)):
            valueCounts = columnDf.value_counts()
            valueCounts.plot.bar()
        else:
            columnDf.hist()
        plt.ylabel('counts')
        plt.xticks(rotation = 90)
        plt.title(f'{columnNames[i]} (column {i})')
    plt.tight_layout(pad = 1.0, w_pad = 1.0, h_pad = 1.0)
    plt.show()
```

In [7]:
```
def plotScatterMatrix(df1, plotSize, textSize):
    df1 = df1.select_dtypes(include =[np.number])

    df1 = df1.dropna('columns')
    df1 = df1[[col for col in df if df[col].nunique() > 1]]
    columnNames = list(df)
    if len(columnNames) > 10:
        columnNames = columnNames[:10]
    df1 = df1[columnNames]
    ax = pd.plotting.scatter_matrix(df1, alpha=0.75, figsize=[plotSize,
 plotSize], diagonal='kde')
    corrs = df1.corr().values
    for i, j in zip(*plt.np.triu_indices_from(ax, k = 1)):
        ax[i, j].annotate('Corr. coef = %.3f' % corrs[i, j], (0.8, 0.2
), xycoords='axes fraction', ha='center', va='center', size=textSize)
    plt.suptitle('Scatter and Density Plot')
    plt.show()
```

In [8]:
```
plotPerColumnDistribution(df, 10, 5)
```



In [9]:
```
plotScatterMatrix(df, 20, 10)
```


Scatter and Density Plot
itching
skin_rash
nodal_skin_eruptions
continuous_sneezing
shivering
chills
joint_pain
stomach_pain
acidity
ulcers_on_tongue
Corr. coef = -0.060
Corr. coef = -0.065
Corr. coef = -0.022
Corr. coef = -0.033
Corr. coef = -0.022
Corr. coef = -0.066
Corr. coef = -0.060
Corr. coef = 0.649
Corr. coef = 0.609
0.0
0.5
1.0
0.00
0.25
0.50
0.75
1.00

In [10]:

```
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_overl
oad  \
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  scurrin
g  \
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_o
oze
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 [11]:
```
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 [12]:

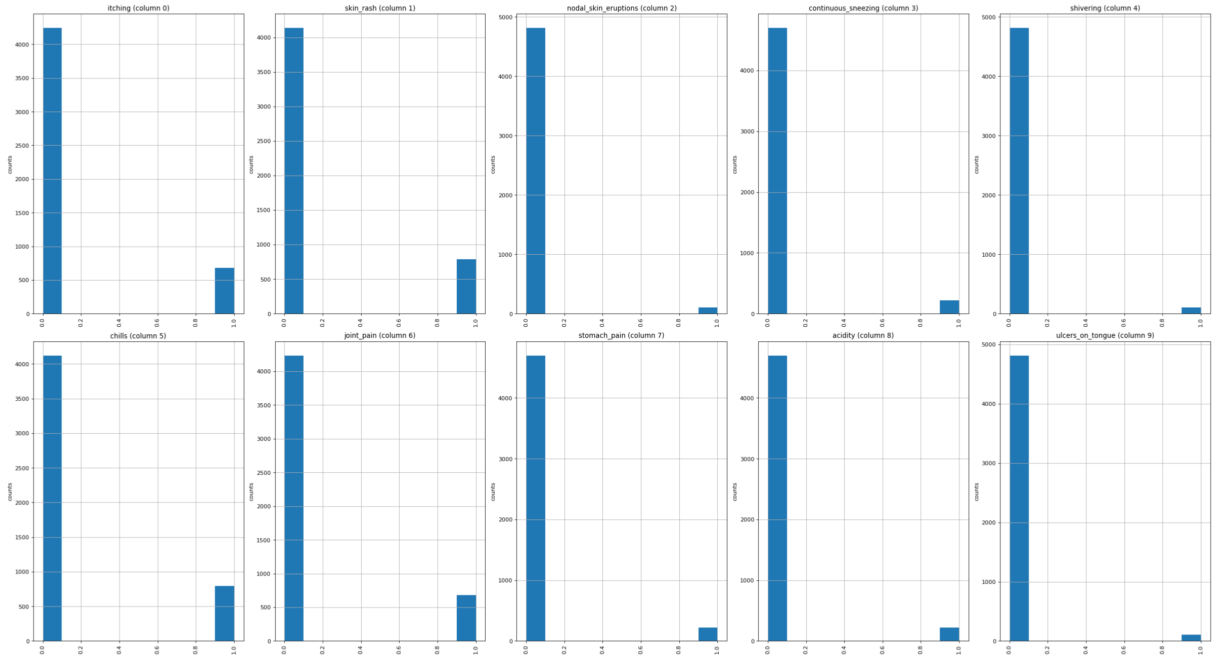
```
tr=pd.read_csv("C:/Users/Damodhar Sai Pesay/Desktop/testing.csv")



tr.replace({'prognosis':{'Fungal infection':0,'Allergy':1,'GERD':2,'Chr
onic cholestasis':3,'Drug Reaction':4,
    'Peptic ulcer diseae':5,'AIDS':6,'Diabetes ':7,'Gastroenteritis':8,
'Bronchial Asthma':9,'Hypertension ':10,
    'Migraine':11,'Cervical spondylosis':12,
    'Paralysis (brain hemorrhage)':13,'Jaundice':14,'Malaria':15,'Chick
en pox':16,'Dengue':17,'Typhoid':18,'hepatitis A':19,
    'Hepatitis B':20,'Hepatitis C':21,'Hepatitis D':22,'Hepatitis E':23
,'Alcoholic hepatitis':24,'Tuberculosis':25,
    'Common Cold':26,'Pneumonia':27,'Dimorphic hemmorhoids(piles)':28,
'Heart attack':29,'Varicose veins':30,'Hypothyroidism':31,
    'Hyperthyroidism':32,'Hypoglycemia':33,'Osteoarthristis':34,'Arthri
tis':35,
    '(vertigo) Paroymsal  Positional Vertigo':36,'Acne':37,'Urinary tra
ct infection':38,'Psoriasis':39,
    'Impetigo':40}},inplace=True)
tr.head()
```

Out[12]:

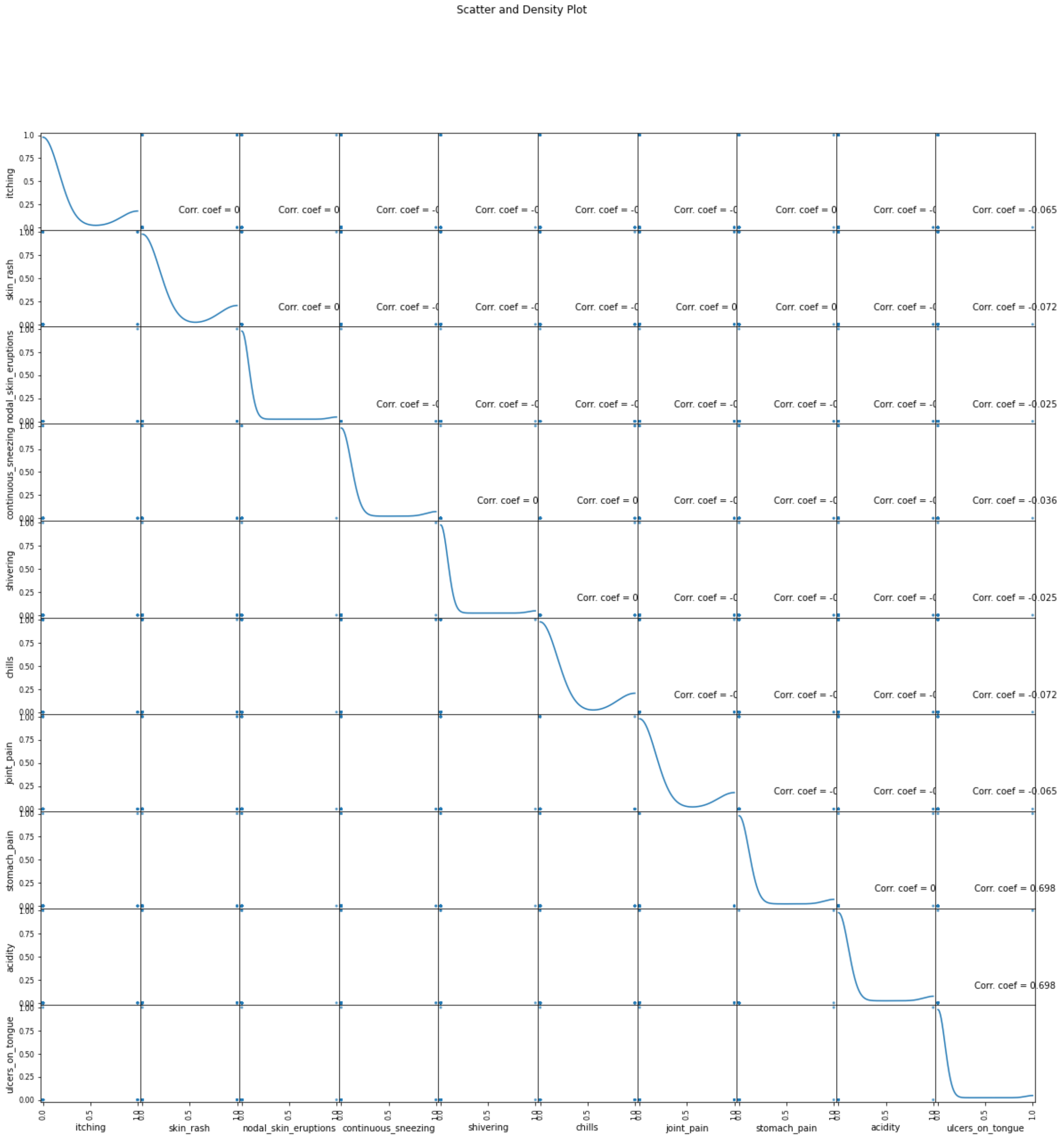
	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	st
0	1	1	1	0	0	0	0	
1	0	0	0	1	1	1	0	
2	0	0	0	0	0	0	0	
3	1	0	0	0	0	0	0	
4	1	1	0	0	0	0	0	

5 rows × 133 columns

```
In [13]: plotPerColumnDistribution(tr, 10, 5)
```



```
In [14]: plotScatterMatrix(tr, 20, 10)
```


Scatter and Density Plot
itching
skin_rash
nodal_skin_eruptions
continuous_sneezing
shivering
chills
joint_pain
stomach_pain
acidity
ulcers_on_tongue
0.00
0.25
0.50
0.75
1.00
0.0
0.5
1.0
Corr. coef = 0
Corr. coef = -0
Corr. coef = -0.065
Corr. coef = -0.072
Corr. coef = -0.025
Corr. coef = -0.036
Corr. coef = -0.025
Corr. coef = -0.072
Corr. coef = -0.065
Corr. coef = 0.698
Corr. coef = 0.698

In [15]:
```
X_test= tr[l1]
y_test = tr[["prognosis"]]
np.ravel(y_test)
print(X_test)
```

```
    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               1          0           0   
4           0             0               0          0           0   
5           0             0               1          0           0   
6           0             0               0          0           0   
7           0             0               0          0           0   
8           0             0               0          1           0   
9           0             0               0          0           0   
10          0             0               0          0           0   
11          0             0               0          0           0   
12          1             0               0          0           0   
13          0             0               0          0           0   
14          0             0               1          0           0   
15          0             0               0          1           0   
16          0             0               0          0           1   
17          1             0               0          0           0   
18          0             1               1          1           0   
19          0             0               1          1           1   
20          0             0               1          0           0   
21          0             0               0          0           0   
22          0             0               1          0           0   
23          0             0               1          0           0   
24          0             0               1          0           0   
25          0             0               0          0           1   
26          0             0               0          0           0   
27          0             0               0          0           0   
28          0             1               0          0           0   
29          0             0               0          0           0   
30          0             0               0          0           0   
31          0             0               0          0           0   
32          0             0               0          1           0   
33          0             0               0          0           0   
```

```
34          0             0               0          0           0
35          0             0               0          0           0
36          0             0               0          0           0
37          0             0               0          0           0
38          0             0               0          0           0
39          0             0               0          0           0
40          0             0               0          0           0

    yellow_urine  yellowing_of_eyes  acute_liver_failure  fluid_overloa
d  \
0              0                  0                    0
0
1              0                  0                    0
0
2              0                  0                    0
0
3              0                  1                    0
0
4              0                  0                    0
0
5              0                  0                    0
0
6              0                  0                    0
0
7              0                  0                    0
0
8              0                  0                    0
0
9              0                  0                    0
0
10             0                  0                    0
0
11             0                  0                    0
0
12             0                  0                    0
0
13             0                  0                    0
0
14             0                  0                    0
```

```
0
15             0                  0                     0
0
16             0                  0                     0
0
17             0                  0                     0
0
18             0                  0                     0
0
19             0                  1                     0
0
20             1                  1                     0
0
21             0                  1                     0
0
22             0                  1                     0
0
23             0                  1                     1
0
24             0                  0                     0
0
25             0                  1                     0
0
26             0                  0                     0
0
27             0                  0                     0
0
28             0                  0                     0
0
29             0                  0                     0
0
30             0                  0                     0
0
31             0                  0                     0
0
32             0                  0                     0
0
33             0                  0                     0
0
```

```
34             0                  0                    0
0
35             0                  0                    0
0
36             0                  0                    0
0
37             0                  0                    0
0
38             0                  0                    0
0
39             0                  0                    0
0
40             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

5                     0  ...                   0           0         0

6                     0  ...                   0           0         0

7                     0  ...                   0           0         0

8                     0  ...                   0           0         0

9                     0  ...                   0           0         0

10                    0  ...                   0           0         0
```

```
11                     0  ...                   0           0         0
12                     0  ...                   0           0         0
13                     0  ...                   0           0         0
14                     0  ...                   0           0         0
15                     0  ...                   0           0         0
16                     0  ...                   0           0         0
17                     0  ...                   0           0         0
18                     0  ...                   0           0         0
19                     0  ...                   0           0         0
20                     0  ...                   0           0         0
21                     0  ...                   0           0         0
22                     0  ...                   0           0         0
23                     0  ...                   0           0         0
24                     1  ...                   0           0         0
25                     0  ...                   0           0         0
26                     0  ...                   0           0         0
27                     0  ...                   0           0         0
28                     0  ...                   0           0         0
29                     0  ...                   0           0         0
30                     0  ...                   0           0         0
```

```
31                    0  ...                   0           0         0
32                    0  ...                   0           0         0
33                    0  ...                   0           0         0
34                    0  ...                   0           0         0
35                    0  ...                   0           0         0
36                    0  ...                   0           0         0
37                    0  ...                   1           1         1
38                    0  ...                   0           0         0
39                    0  ...                   0           0         0
40                    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
5              0                    0                     0
6              0                    0                     0
7              0                    0                     0
8              0                    0                     0
9              0                    0                     0
10             0                    0                     0
11             0                    0                     0
12             0                    0                     0
13             0                    0                     0
14             0                    0                     0
15             0                    0                     0
```

```
16             0                    0                     0
17             0                    0                     0
18             0                    0                     0
19             0                    0                     0
20             0                    0                     0
21             0                    0                     0
22             0                    0                     0
23             0                    0                     0
24             0                    0                     0
25             0                    0                     0
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_ooz
e
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
5                    0        0                     0
```

```
0
6                    0        0                    0
0
7                    0        0                    0
0
8                    0        0                    0
0
9                    0        0                    0
0
10                   0        0                    0
0
11                   0        0                    0
0
12                   0        0                    0
0
13                   0        0                    0
0
14                   0        0                    0
0
15                   0        0                    0
0
16                   0        0                    0
0
17                   0        0                    0
0
18                   0        0                    0
0
19                   0        0                    0
0
20                   0        0                    0
0
21                   0        0                    0
0
22                   0        0                    0
0
23                   0        0                    0
0
24                   0        0                    0
0
```

```
25                   0        0                     0
0
26                   0        0                     0
0
27                   0        0                     0
0
28                   0        0                     0
0
29                   0        0                     0
0
30                   0        0                     0
0
31                   0        0                     0
0
32                   0        0                     0
0
33                   0        0                     0
0
34                   0        0                     0
0
35                   0        0                     0
0
36                   0        0                     0
0
37                   0        0                     0
0
38                   0        0                     0
0
39                   1        0                     0
0
40                   0        1                     1
1

[41 rows x 95 columns]
```

In [16]:
```
print(y_test)
```

```
    prognosis
0           0
```

```
39          39
40          40
```

To build the precision of the model we have used four algorithms.

1) Decision Tree algorithm 2) Random Forest algorithm 3) KNearestNeighbour algorithm 4) Naive Bayes algorithm

In [17]:

```
def scatterplt(disea):
    x = ((DF.loc[disea]).sum())
    x.drop(x[x==0].index,inplace=True)
    print(x.values)
    y = x.keys()
    print(len(x))
    print(len(y))
    plt.title(disea)
    plt.scatter(y,x.values)
    plt.show()

def scatterinp(sym1,sym2,sym3,sym4,sym5):
    x = [sym1,sym2,sym3,sym4,sym5]
    y = [0,0,0,0,0]
    if(sym1!='Select Here'):
        y[0]=1
    if(sym2!='Select Here'):
        y[1]=1
    if(sym3!='Select Here'):
        y[2]=1
    if(sym4!='Select Here'):
        y[3]=1
    if(sym5!='Select Here'):
        y[4]=1
    print(x)
    print(y)
    plt.scatter(x,y)
    plt.show()
```

Decision Tree Algorithm

In [18]:

```
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 He
re")):
        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,confusion_mat
rix,accuracy_score
        y_pred=clf3.predict(X_test)
        print("Decision Tree")
        print("Accuracy")
        print(accuracy_score(y_test, y_pred))
        print(accuracy_score(y_test, y_pred,normalize=False))
        print("Confusion matrix")
        conf_matrix=confusion_matrix(y_test,y_pred)
        print(conf_matrix)

        psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Sympt
om4.get(),Symptom5.get()]

        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")

        import sqlite3
        conn = sqlite3.connect('database.db')
        c = conn.cursor()
        c.execute("CREATE TABLE IF NOT EXISTS DecisionTree(Name StringV
ar,Symtom1 StringVar,Symtom2 StringVar,Symtom3 StringVar,Symtom4 TEXT,S
ymtom5 TEXT,Disease StringVar)")
        c.execute("INSERT INTO DecisionTree(Name,Symtom1,Symtom2,Symtom
3,Symtom4,Symtom5,Disease) VALUES(?,?,?,?,?,?,?)",(NameEn.get(),Symptom
1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get(),pre
d1.get()))
        conn.commit()
        c.close()
        conn.close()


        scatterinp(Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom
4.get(),Symptom5.get())
        scatterplt(pred1.get())
```

Random Forest Algorithm

In [19]:

```
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))

        # calculating accuracy
        from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
        y_pred=clf4.predict(X_test)
        print("Random Forest")
        print("Accuracy")
        print(accuracy_score(y_test, y_pred))
        print(accuracy_score(y_test, y_pred,normalize=False))
        print("Confusion matrix")
        conf_matrix=confusion_matrix(y_test,y_pred)
        print(conf_matrix)

        psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Sympt
om4.get(),Symptom5.get()]

        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")

        import sqlite3
        conn = sqlite3.connect('database.db')
        c = conn.cursor()
        c.execute("CREATE TABLE IF NOT EXISTS RandomForest(Name StringV
ar,Symtom1 StringVar,Symtom2 StringVar,Symtom3 StringVar,Symtom4 TEXT,S
ymtom5 TEXT,Disease StringVar)")
        c.execute("INSERT INTO RandomForest(Name,Symtom1,Symtom2,Symtom
3,Symtom4,Symtom5,Disease) VALUES(?,?,?,?,?,?,?)",(NameEn.get(),Symptom
1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get(),pre
d2.get()))
        conn.commit()
        c.close()
        conn.close()

        scatterplt(pred2.get())
```

KNearestNeighbour Algorithm

In [20]:

```
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,confusion_matrix,accuracy_score
        y_pred=knn.predict(X_test)
        print("kNearest Neighbour")
        print("Accuracy")
        print(accuracy_score(y_test, y_pred))
        print(accuracy_score(y_test, y_pred,normalize=False))
        print("Confusion matrix")
        conf_matrix=confusion_matrix(y_test,y_pred)
        print(conf_matrix)

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

        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")

        import sqlite3
        conn = sqlite3.connect('database.db')
        c = conn.cursor()
        c.execute("CREATE TABLE IF NOT EXISTS KNearestNeighbour(Name St
ringVar,Symtom1 StringVar,Symtom2 StringVar,Symtom3 StringVar,Symtom4 T
EXT,Symtom5 TEXT,Disease StringVar)")
        c.execute("INSERT INTO KNearestNeighbour(Name,Symtom1,Symtom2,S
ymtom3,Symtom4,Symtom5,Disease) VALUES(?,?,?,?,?,?,?)",(NameEn.get(),Sy
mptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get
(),pred4.get()))
        conn.commit()
        c.close()
        conn.close()


        scatterplt(pred4.get())
```

Naive Bayes Algorithm

In [21]:

```
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,confusion_matrix,accuracy_score
        y_pred=gnb.predict(X_test)
        print("Naive Bayes")
        print("Accuracy")
        print(accuracy_score(y_test, y_pred))
        print(accuracy_score(y_test, y_pred,normalize=False))
        print("Confusion matrix")
        conf_matrix=confusion_matrix(y_test,y_pred)
        print(conf_matrix)

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

        import sqlite3
        conn = sqlite3.connect('database.db')
        c = conn.cursor()
        c.execute("CREATE TABLE IF NOT EXISTS NaiveBayes(Name StringVa
r,Symtom1 StringVar,Symtom2 StringVar,Symtom3 StringVar,Symtom4 TEXT,Sy
mtom5 TEXT,Disease StringVar)")
        c.execute("INSERT INTO NaiveBayes(Name,Symtom1,Symtom2,Symtom3,
Symtom4,Symtom5,Disease) VALUES(?,?,?,?,?,?,?)",(NameEn.get(),Symptom1.
get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get(),pred3
.get()))
        conn.commit()
        c.close()
        conn.close()

        scatterplt(pred3.get())
```

Building Graphical User Interface

In [22]:

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

Out[22]: ''

In [23]:

```
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 [24]:

```
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 [25]:

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

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

In [26]:

```
#Headings for the GUI written at the top of GUI
w2 = Label(root, justify=LEFT, text="Disease Predictor using Machine Le
arning", fg="Red", bg="Ivory")
w2.config(font=("Times",30,"bold italic"))
w2.grid(row=1, column=0, columnspan=2, padx=100)
w2 = Label(root, justify=LEFT, text="Contributors: Damodhar,Surya,Phan
i,Karthik", fg="Pink", bg="Ivory")
w2.config(font=("Times",30,"bold italic"))
w2.grid(row=2, column=0, columnspan=2, padx=100)
```

In [27]:

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

In [28]:

```
#Creating Labels for the symtoms
S1Lb = Label(root, text="Symptom 1 *", 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 *", 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 [29]:

```
#Labels for the different algorithms
lrLb = Label(root, text="DecisionTree", fg="white", bg="red", width = 2
0)
lrLb.config(font=("Times",15,"bold italic"))
lrLb.grid(row=15, column=0, pady=10,sticky=W)

destreeLb = Label(root, text="RandomForest", fg="Red", bg="Orange", wid
th = 20)
destreeLb.config(font=("Times",15,"bold italic"))
destreeLb.grid(row=17, column=0, pady=10, sticky=W)

ranfLb = Label(root, text="NaiveBayes", fg="White", bg="green", width =
 20)
ranfLb.config(font=("Times",15,"bold italic"))
ranfLb.grid(row=19, column=0, pady=10, sticky=W)

knnLb = Label(root, text="kNearestNeighbour", fg="Red", bg="Sky Blue",
width = 20)
knnLb.config(font=("Times",15,"bold italic"))
knnLb.grid(row=21, column=0, pady=10, sticky=W)
OPTIONS = sorted(l1)
```

In [30]:

```
#Taking name as input from user
NameEn = Entry(root, textvariable=Name)
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 [31]:

```
#Buttons for predicting the disease using different algorithms
dst = Button(root, text="Prediction 1", command=DecisionTree,bg="Red",f
g="yellow")
dst.config(font=("Times",15,"bold italic"))
dst.grid(row=6, column=3,padx=10)

rnf = Button(root, text="Prediction 2", command=randomforest,bg="Light
 green",fg="red")
rnf.config(font=("Times",15,"bold italic"))
rnf.grid(row=7, column=3,padx=10)

lr = Button(root, text="Prediction 3", command=NaiveBayes,bg="Blue",fg=
"white")
lr.config(font=("Times",15,"bold italic"))
lr.grid(row=8, column=3,padx=10)

kn = Button(root, text="Prediction 4", command=KNN,bg="sky blue",fg="re
d")
kn.config(font=("Times",15,"bold italic"))
kn.grid(row=9, column=3,padx=10)

rs = Button(root,text="Reset Inputs", command=Reset,bg="yellow",fg="pur
ple",width=15)
rs.config(font=("Times",15,"bold italic"))
rs.grid(row=10,column=3,padx=10)

ex = Button(root,text="Exit System", command=Exit,bg="yellow",fg="purpl
e",width=15)
ex.config(font=("Times",15,"bold italic"))
ex.grid(row=11,column=3,padx=10)
```

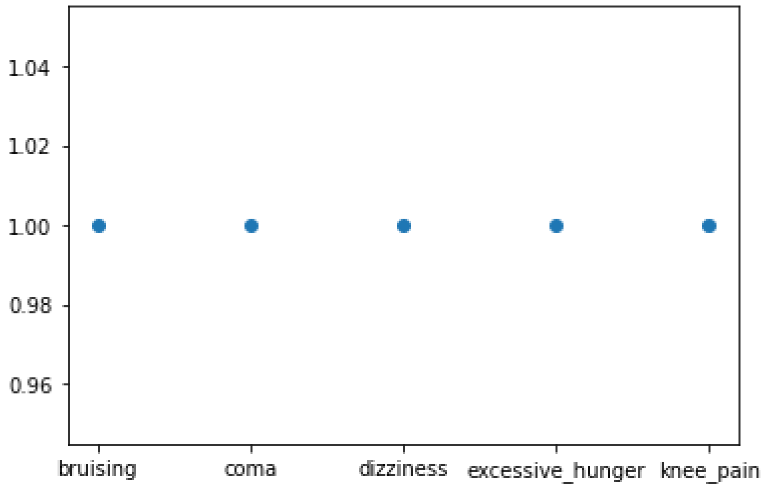
In [32]:

```
#Showing the output of different aldorithms
t1=Label(root,font=("Times",15,"bold italic"),text="Decision Tree",heig
ht=1,bg="Light green"
         ,width=40,fg="red",textvariable=pred1,relief="sunken").grid(ro
w=15, column=1, padx=10)

t2=Label(root,font=("Times",15,"bold italic"),text="Random Forest",heig
ht=1,bg="Purple"
         ,width=40,fg="white",textvariable=pred2,relief="sunken").grid(
row=17, column=1, padx=10)

t3=Label(root,font=("Times",15,"bold italic"),text="Naive Bayes",height
=1,bg="red"
         ,width=40,fg="orange",textvariable=pred3,relief="sunken").grid
(row=19, column=1, padx=10)

t4=Label(root,font=("Times",15,"bold italic"),text="kNearest Neighbour"
,height=1,bg="Blue"
         ,width=40,fg="yellow",textvariable=pred4,relief="sunken").grid
(row=21, column=1, padx=10)
```

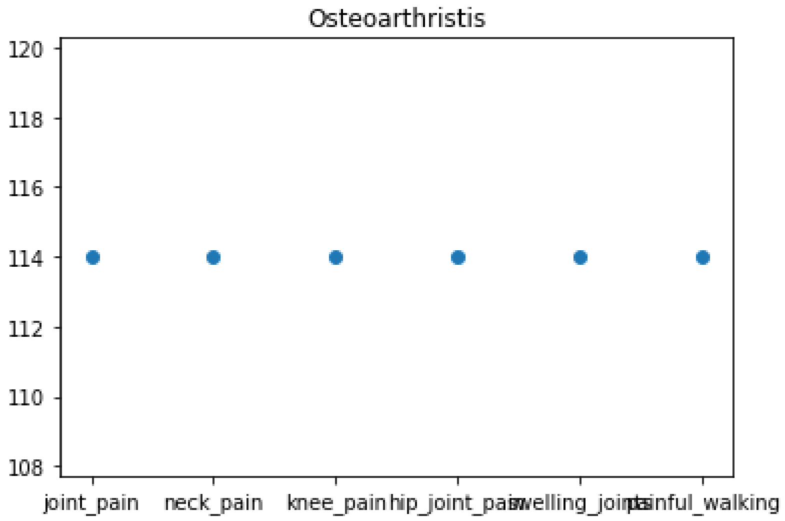
In [ ]:

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

```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['bruising', 'coma', 'dizziness', 'excessive_hunger', 'knee_pain']
[1, 1, 1, 1, 1]
```

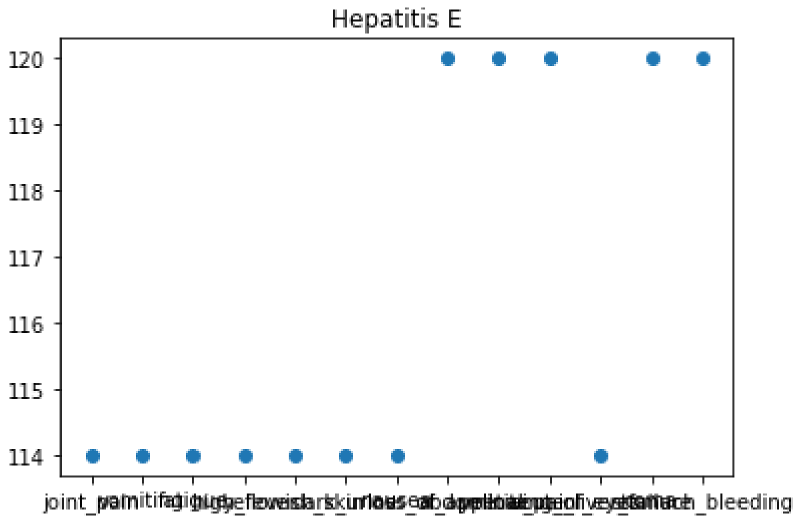


```
[114 114 114 114 114 114]
6
6
```

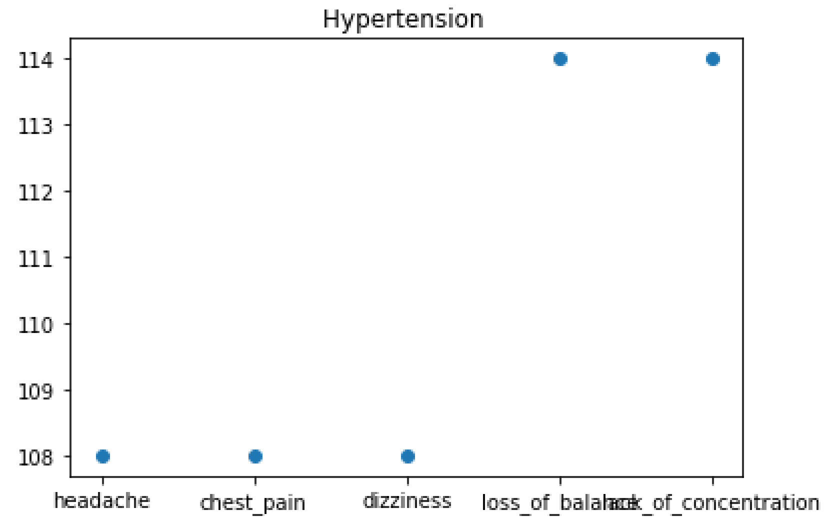


```
Random Forest
```

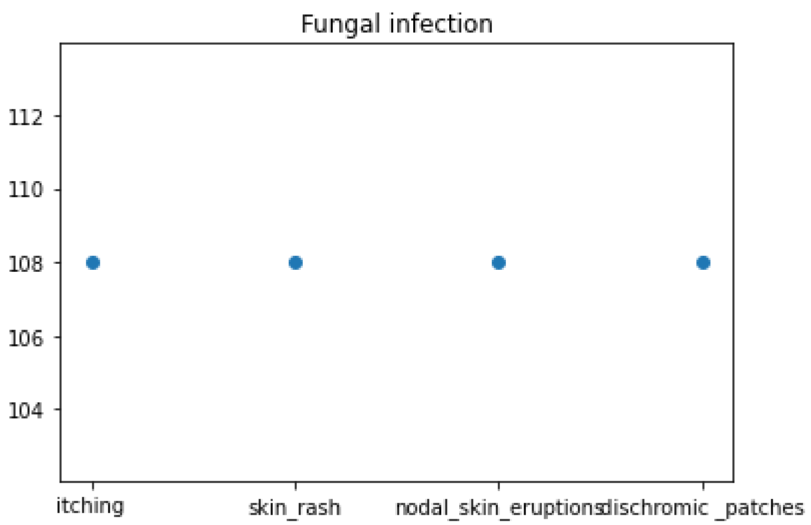
```
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[114 114 114 114 114 114 114 120 120 120 114 120 120]
13
13
```



```
Naive Bayes
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[108 108 108 114 114]
5
5
```



```
kNearest Neighbour
Accuracy
0.926829268292683
38

Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[108 108 108 108]
4
4
```


Fungal infection
112
110
108
106
104
itching
skin_rash
nodal_skin_eruptions
dischromic _patches

```
In [ ]:
```