In [74]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
```

In [4]:
```
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

In [5]:
```
train_df=pd.read_csv("C:\\Users\\damma\\Downloads\\Test.csv")
df=train_df.copy()
```

In [6]:
```
test_df=pd.read_csv("C:\\Users\\damma\\Downloads\\Training.csv")
df=test_df.copy()
```

Exploratory Data Analysis

In [7]:
```
test_df.head()
```

Out[7]:

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	scurring	skin_
0	1	1	1	0	0	0	0	0	0	0	...	0	
1	0	1	1	0	0	0	0	0	0	0	...	0	
2	1	0	1	0	0	0	0	0	0	0	...	0	
3	1	1	0	0	0	0	0	0	0	0	...	0	
4	1	1	1	0	0	0	0	0	0	0	...	0	

5 rows × 134 columns

In [13]:
```
test_df.tail()
```

Out[13]:

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	scurring	s
4915	0	0	0	0	0	0	0	0	0	0	...	0	
4916	0	1	0	0	0	0	0	0	0	0	...	1	
4917	0	0	0	0	0	0	0	0	0	0	...	0	
4918	0	1	0	0	0	0	1	0	0	0	...	0	
4919	0	1	0	0	0	0	0	0	0	0	...	0	

5 rows × 134 columns

In [8]:
```
train_df.head()
```

Out[8]:

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	blackheads	sc
0	1	1	1	0	0	0	0	0	0	0	...	0	
1	0	0	0	1	1	1	0	0	0	0	...	0	
2	0	0	0	0	0	0	0	1	1	1	...	0	
3	1	0	0	0	0	0	0	0	0	0	...	0	
4	1	1	0	0	0	0	0	1	0	0	...	0	

5 rows × 133 columns

In [14]:
```
train_df.tail()
```

Out[14]:

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	blackheads	s
37	0	1	0	0	0	0	0	0	0	0	...	1	
38	0	0	0	0	0	0	0	0	0	0	...	0	
39	0	1	0	0	0	0	1	0	0	0	...	0	
40	0	1	0	0	0	0	0	0	0	0	...	0	
41	1	1	0	0	0	0	0	0	0	0	...	0	

5 rows × 133 columns

In [9]:
```
train_df.describe()
```

Out[9]:

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...
count	42.000000	42.000000	42.000000	42.000000	42.000000	42.000000	42.000000	42.000000	42.000000	42.000000	...
mean	0.166667	0.190476	0.023810	0.047619	0.023810	0.166667	0.142857	0.047619	0.047619	0.023810	...
std	0.377195	0.397437	0.154303	0.215540	0.154303	0.377195	0.354169	0.215540	0.215540	0.154303	...
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	...

8 rows × 132 columns

In [10]:
```
train_df.describe().T
```

Out[10]:

	count	mean	std	min	25%	50%	75%	max
itching	42.0	0.166667	0.377195	0.0	0.0	0.0	0.0	1.0
skin_rash	42.0	0.190476	0.397437	0.0	0.0	0.0	0.0	1.0
nodal_skin_eruptions	42.0	0.023810	0.154303	0.0	0.0	0.0	0.0	1.0
continuous_sneezing	42.0	0.047619	0.215540	0.0	0.0	0.0	0.0	1.0
shivering	42.0	0.023810	0.154303	0.0	0.0	0.0	0.0	1.0
...	...	...	...	...	...	...	...	...
small_dents_in_nails	42.0	0.023810	0.154303	0.0	0.0	0.0	0.0	1.0
inflammatory_nails	42.0	0.023810	0.154303	0.0	0.0	0.0	0.0	1.0
blister	42.0	0.023810	0.154303	0.0	0.0	0.0	0.0	1.0
red_sore_around_nose	42.0	0.047619	0.215540	0.0	0.0	0.0	0.0	1.0
yellow_crust_ooze	42.0	0.023810	0.154303	0.0	0.0	0.0	0.0	1.0

132 rows × 8 columns

In [11]:
```
test_df.describe()
```

Out[11]:

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on
count	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	492
mean	0.137805	0.159756	0.021951	0.045122	0.021951	0.162195	0.139024	0.045122	0.045122	
std	0.344730	0.366417	0.146539	0.207593	0.146539	0.368667	0.346007	0.207593	0.207593	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	

8 rows × 133 columns

In [12]:
```
test_df.describe().T
```

Out[12]:

	count	mean	std	min	25%	50%	75%	max
itching	4920.0	0.137805	0.344730	0.0	0.0	0.0	0.0	1.0
skin_rash	4920.0	0.159756	0.366417	0.0	0.0	0.0	0.0	1.0
nodal_skin_eruptions	4920.0	0.021951	0.146539	0.0	0.0	0.0	0.0	1.0
continuous_sneezing	4920.0	0.045122	0.207593	0.0	0.0	0.0	0.0	1.0
shivering	4920.0	0.021951	0.146539	0.0	0.0	0.0	0.0	1.0
...	...	...	...	...	...	...	...	...
inflammatory_nails	4920.0	0.023171	0.150461	0.0	0.0	0.0	0.0	1.0
blister	4920.0	0.023171	0.150461	0.0	0.0	0.0	0.0	1.0
red_sore_around_nose	4920.0	0.023171	0.150461	0.0	0.0	0.0	0.0	1.0
yellow_crust_ooze	4920.0	0.023171	0.150461	0.0	0.0	0.0	0.0	1.0
Unnamed: 133	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN

133 rows × 8 columns

In [15]:
```
train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42 entries, 0 to 41
Columns: 133 entries, itching to prognosis
dtypes: int64(132), object(1)
memory usage: 43.8+ KB
```

In [16]:
```
test_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4920 entries, 0 to 4919
Columns: 134 entries, itching to Unnamed: 133
dtypes: float64(1), int64(132), object(1)
memory usage: 5.0+ MB
```

In [18]:
```
print(test_df.shape)
```

```
(4920, 134)
```

In [17]:
```
print(train_df.shape)
```

```
(42, 133)
```

In [17]:
```
train_df.columns
```

Out[17]:
```
Index(['itching', 'skin_rash', 'nodal_skin_eruptions', 'continuous_sneezing',
       'shivering', 'chills', 'joint_pain', 'stomach_pain', 'acidity',
       'ulcers_on_tongue',
       ...
       'blackheads', 'scurring', 'skin_peeling', 'silver_like_dusting',
       'small_dents_in_nails', 'inflammatory_nails', 'blister',
       'red_sore_around_nose', 'yellow_crust_ooze', 'prognosis'],
      dtype='object', length=133)
```

In [19]:
```
test_df.columns
```

Out[19]:
```
Index(['itching', 'skin_rash', 'nodal_skin_eruptions', 'continuous_sneezing',
       'shivering', 'chills', 'joint_pain', 'stomach_pain', 'acidity',
       'ulcers_on_tongue',
       ...
       'scurring', 'skin_peeling', 'silver_like_dusting',
       'small_dents_in_nails', 'inflammatory_nails', 'blister',
       'red_sore_around_nose', 'yellow_crust_ooze', 'prognosis',
       'Unnamed: 133'],
      dtype='object', length=134)
```

In [55]:
```
train_df.isna().sum()
```

Out[55]:
```
itching                0
skin_rash              0
nodal_skin_eruptions   0
continuous_sneezing    0
shivering              0
                      ..
inflammatory_nails     0
blister                0
red_sore_around_nose   0
yellow_crust_ooze      0
prognosis              0
Length: 133, dtype: int64
```

In [22]:
```
test_df.isna().sum()
```

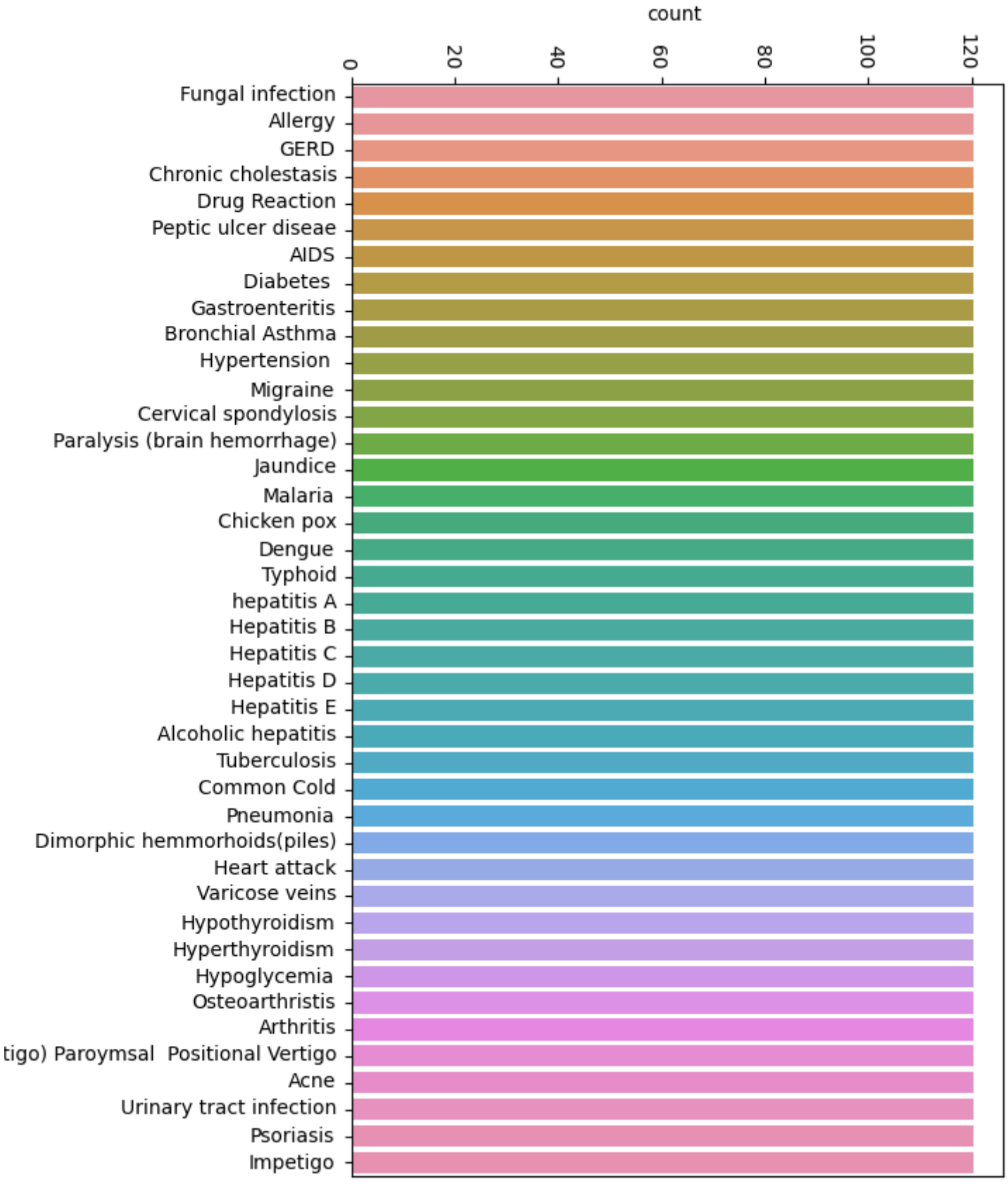
Out[22]:
```
itching                    0
skin_rash                  0
nodal_skin_eruptions       0
continuous_sneezing        0
shivering                  0
                         ...
blister                    0
red_sore_around_nose       0
yellow_crust_ooze          0
prognosis                  0
Unnamed: 133            4920
Length: 134, dtype: int64
```

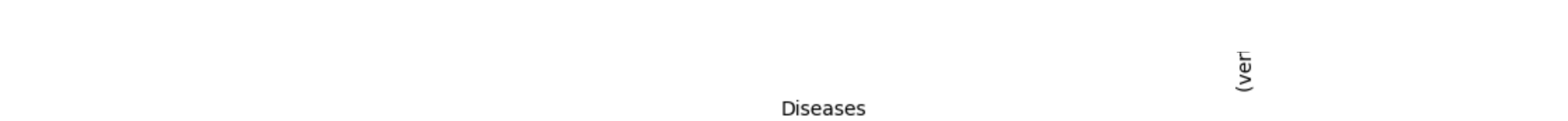
In [32]:
```
df['prognosis'].value_counts()
```

Out[32]:

```
Fungal infection                          120
Hepatitis C                               120
Hepatitis E                               120
Alcoholic hepatitis                       120
Tuberculosis                              120
Common Cold                               120
Pneumonia                                 120
Dimorphic hemmorhoids(piles)              120
Heart attack                              120
Varicose veins                            120
Hypothyroidism                            120
Hyperthyroidism                           120
Hypoglycemia                              120
Osteoarthristis                           120
Arthritis                                 120
(vertigo) Paroymsal  Positional Vertigo   120
Acne                                      120
Urinary tract infection                   120
Psoriasis                                 120
Hepatitis D                               120
Hepatitis B                               120
Allergy                                   120
hepatitis A                               120
GERD                                      120
Chronic cholestasis                       120
Drug Reaction                             120
Peptic ulcer diseae                       120
AIDS                                      120
Diabetes                                  120
Gastroenteritis                           120
Bronchial Asthma                          120
Hypertension                              120
Migraine                                  120
Cervical spondylosis                      120
Paralysis (brain hemorrhage)              120
Jaundice                                  120
Malaria                                   120
Chicken pox                               120
Dengue                                    120
Typhoid                                   120
Impetigo                                  120
Name: prognosis, dtype: int64
```

In [31]:
```
plt.figure(figsize= (10,6))
sns.countplot(data = df, x= 'prognosis')
plt.xticks(rotation=90)
plt.xlabel('Diseases');
```


count
0
20
40
60
80
100
120
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
tigo) Paroymsal Positional Vertigo
Acne
Urinary tract infection
Psoriasis
Impetigo



In [40]:
```
mylabels = list(y.unique())
mylabels
```

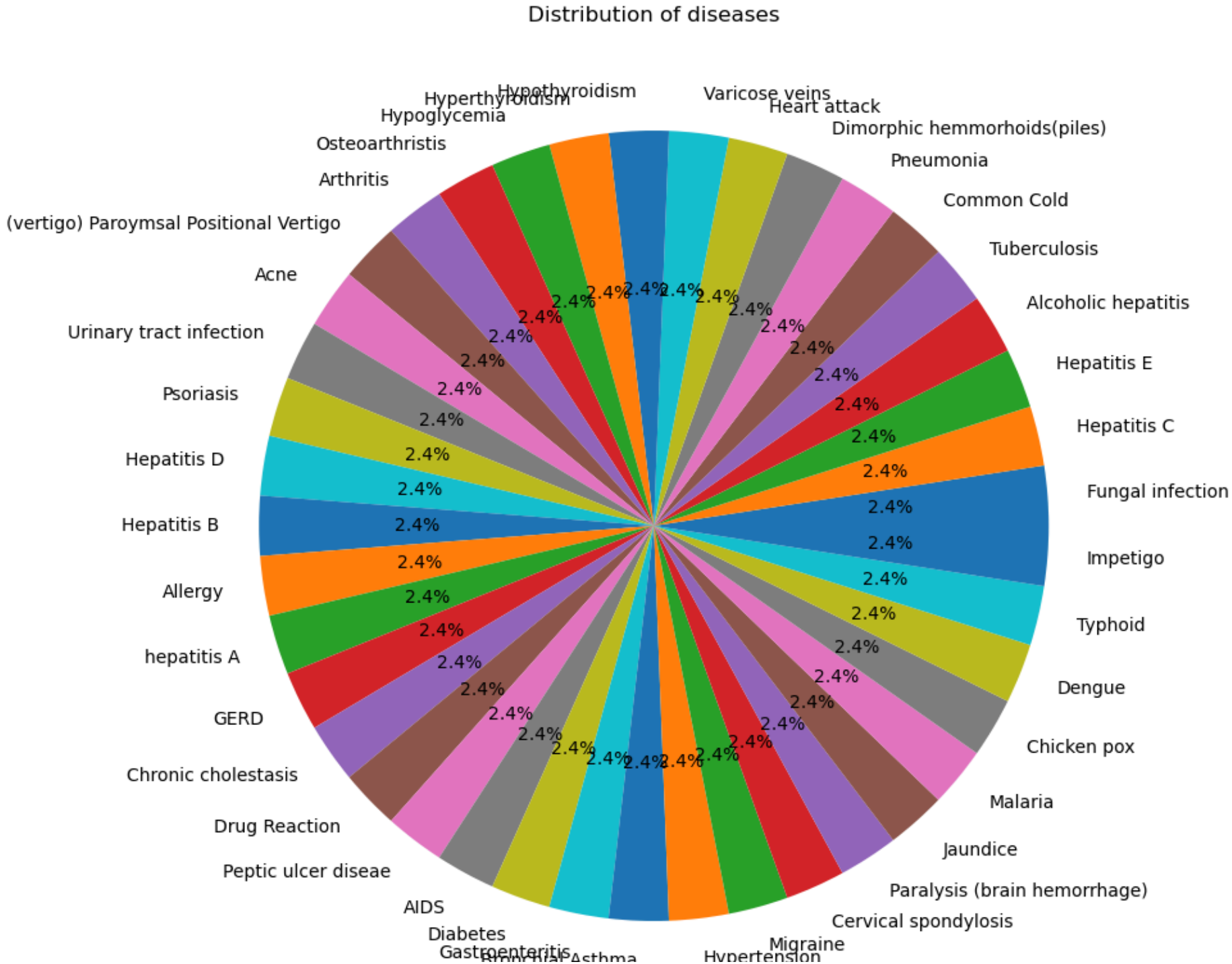
Out[40]:
```
[120]
```

In [43]:
```
y = df['prognosis'].value_counts()
```

In [46]:
```
diseases = ["Fungal infection", "Hepatitis C", "Hepatitis E", "Alcoholic hepatitis", "Tuberculosis", "Common Cold", "Pneumonia",

cases = [120] * len(diseases)

plt.figure(figsize=(10, 10))
plt.pie(cases, labels=diseases, autopct='%1.1f%%')
plt.title("Distribution of diseases")
plt.show()
```


Distribution of diseases
Hypothyroidism
Varicose veins
Hyperthyroidism
Heart attack
Hypoglycemia
Dimorphic hemmorhoids(piles)
Osteoarthristis
Pneumonia
Arthritis
Common Cold
(vertigo) Paroymsal Positional Vertigo
Tuberculosis
Acne
Alcoholic hepatitis
Urinary tract infection
Hepatitis E
Psoriasis
Hepatitis C
Hepatitis D
Fungal infection
Hepatitis B
Impetigo
Allergy
Typhoid
hepatitis A
Dengue
GERD
Chicken pox
Chronic cholestasis
Malaria
Drug Reaction
Jaundice
Peptic ulcer diseae
Paralysis (brain hemorrhage)
AIDS
Cervical spondylosis
Diabetes
Migraine
Gastroenteritis
Hypertension
2.4%

Bronchial Asthma Hypertension

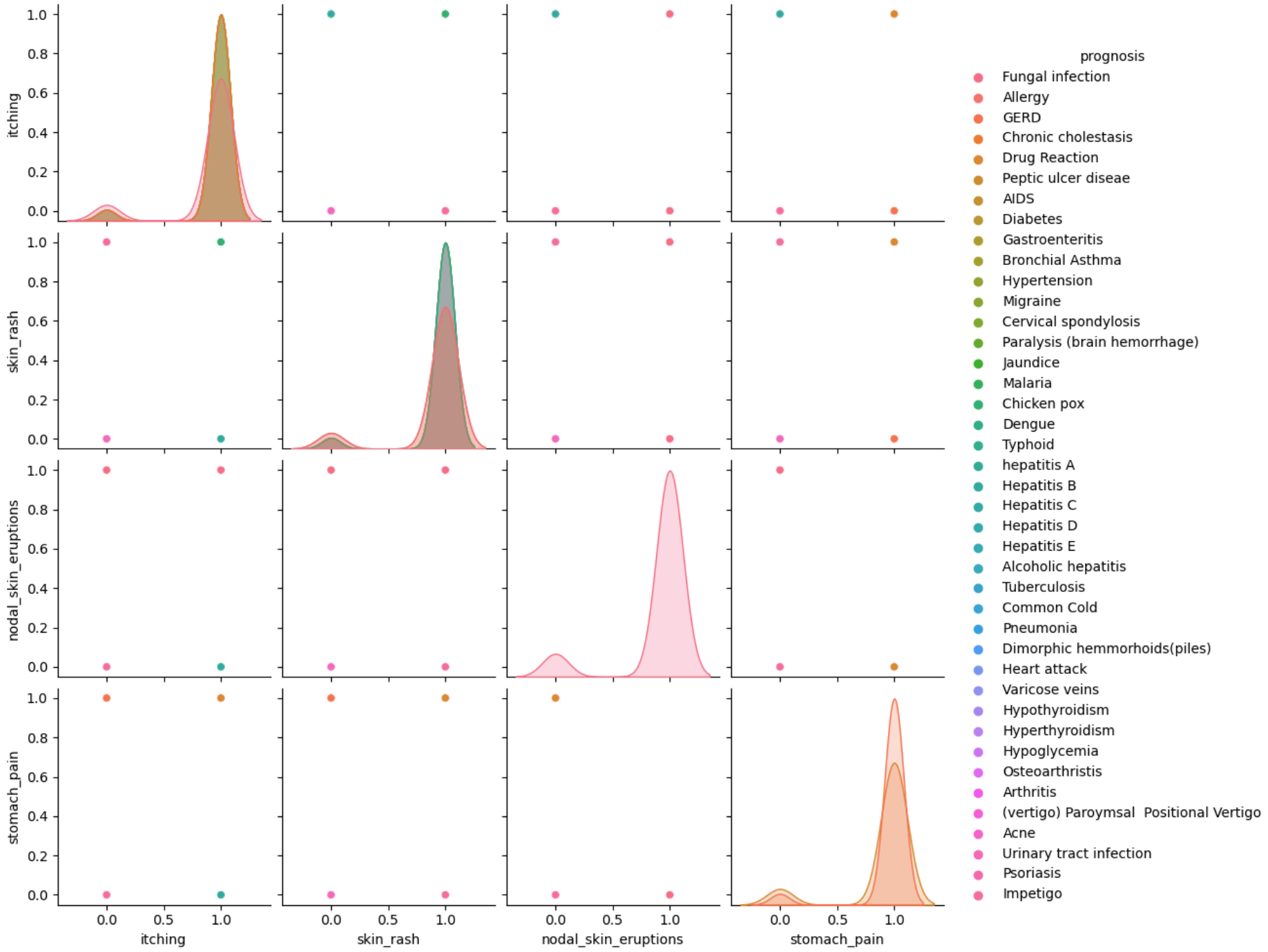
In [58]:
```
test_df=test_df.drop(["Unnamed: 133"],axis=1)
```

In [63]:
```
selected_columns = ['itching', 'skin_rash', 'nodal_skin_eruptions', 'stomach_pain', 'prognosis']

test_df_selected = test_df[selected_columns]

sns.pairplot(test_df_selected, hue='prognosis')

plt.show()
```


itching
skin_rash
nodal_skin_eruptions
stomach_pain
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
prognosis
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 [64]:
```
num = ['itching', 'skin_rash', 'nodal_skin_eruptions', 'continuous_sneezing', 'shivering', 'chills', 'joint_pain',
       'stomach_pain', 'acidity', 'ulcers_on_tongue', 'muscle_wasting', 'vomiting', 'burning_micturition',
       'spotting_ urination', 'fatigue', 'weight_gain', 'anxiety', 'cold_hands_and_feets', 'mood_swings',
       'weight_loss', 'restlessness', 'lethargy', 'patches_in_throat', 'irregular_sugar_level', 'cough',
       'high_fever', 'sunken_eyes', 'breathlessness', 'sweating', 'dehydration', 'indigestion', 'headache',
       'yellowish_skin', 'dark_urine', 'nausea', 'loss_of_appetite', 'pain_behind_the_eyes', '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_v
       'runny_nose', 'congestion', 'chest_pain', 'weakness_in_limbs', 'fast_heart_rate',
       'pain_during_bowel_movements', 'pain_in_anal_region', 'bloody_stool', 'irritation_in_anus', 'neck_pain',
       'dizziness', 'cramps', 'bruising', 'obesity', 'swollen_legs', 'swollen_blood_vessels', 'puffy_face_and_eyes',
       'enlarged_thyroid', 'brittle_nails', 'swollen_extremeties', 'excessive_hunger', 'extra_marital_contacts',
       'drying_and_tingling_lips', 'slurred_speech', 'knee_pain', 'hip_joint_pain', 'muscle_weakness', 'stiff_neck',
       'swelling_joints', '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', 'pas
       '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', 'receiving_blood_transfusion', 'receiving_unsterile_injections', 'coma',
       'stomach_bleeding', 'distention_of_abdomen', 'history_of_alcohol_consumption', 'fluid_overload.1',
       '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', '
```

In [65]:
```
#correlation
df[num].corr()
```

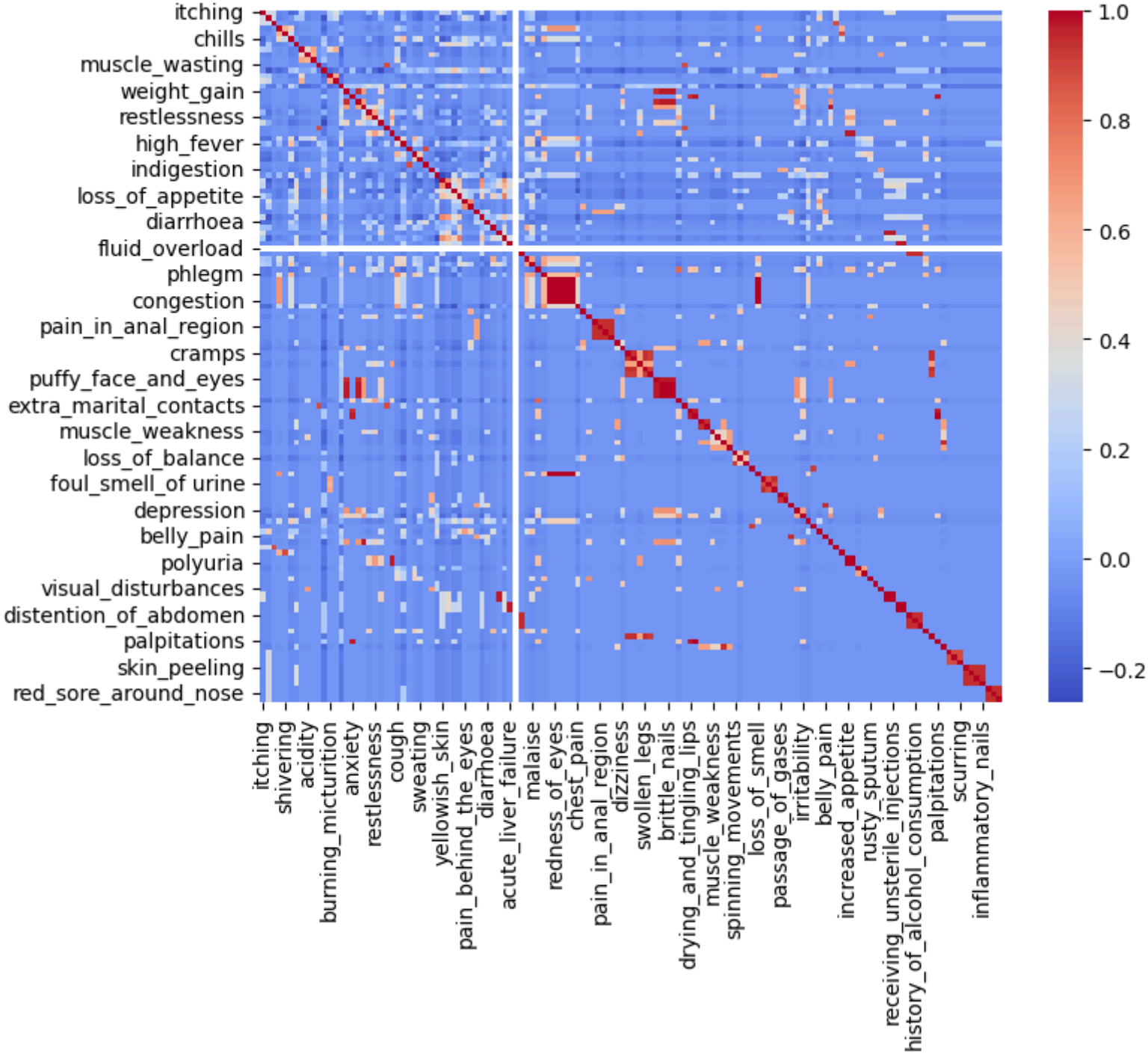
Out[65]:

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_o
itching	1.000000	0.318158	0.326439	-0.086906	-0.059893	-0.175905	-0.160650	0.202850	-0.086906	
skin_rash	0.318158	1.000000	0.298143	-0.094786	-0.065324	-0.029324	0.171134	0.161784	-0.094786	
nodal_skin_eruptions	0.326439	0.298143	1.000000	-0.032566	-0.022444	-0.065917	-0.060200	-0.032566	-0.032566	
continuous_sneezing	-0.086906	-0.094786	-0.032566	1.000000	0.608981	0.446238	-0.087351	-0.047254	-0.047254	
shivering	-0.059893	-0.065324	-0.022444	0.608981	1.000000	0.295332	-0.060200	-0.032566	-0.032566	
...	...	...	...	...	...	...	...	...	...	
small_dents_in_nails	-0.061573	0.331087	-0.023073	-0.033480	-0.023073	-0.067765	0.359845	-0.033480	-0.033480	
inflammatory_nails	-0.061573	0.331087	-0.023073	-0.033480	-0.023073	-0.067765	0.359845	-0.033480	-0.033480	
blister	-0.061573	0.331087	-0.023073	-0.033480	-0.023073	-0.067765	-0.061889	-0.033480	-0.033480	
red_sore_around_nose	-0.061573	0.331087	-0.023073	-0.033480	-0.023073	-0.067765	-0.061889	-0.033480	-0.033480	
yellow_crust_ooze	-0.061573	0.331087	-0.023073	-0.033480	-0.023073	-0.067765	-0.061889	-0.033480	-0.033480	

132 rows × 132 columns

In [67]:
```
#heatmap
plt.figure(figsize=(8,6))
sns.heatmap(data=df[num].corr(),cmap='coolwarm')
```

Out[67]:
```
<Axes: >
```


itching
chills
muscle_wasting
weight_gain
restlessness
high_fever
indigestion
loss_of_appetite
diarrhoea
fluid_overload
phlegm
congestion
pain_in_anal_region
cramps
puffy_face_and_eyes
extra_marital_contacts
muscle_weakness
loss_of_balance
foul_smell_of urine
depression
belly_pain
polyuria
visual_disturbances
distention_of_abdomen
palpitations
skin_peeling
red_sore_around_nose
itching
shivering
acidity
burning_micturition
anxiety
restlessness
cough
sweating
yellowish_skin
pain_behind_the_eyes
diarrhoea
acute_liver_failure
malaise
redness_of_eyes
chest_pain
pain_in_anal_region
dizziness
swollen_legs
brittle_nails
drying_and_tingling_lips
muscle_weakness
spinning_movements
loss_of_smell
passage_of_gases
irritability
belly_pain
increased_appetite
rusty_sputum
receiving_unsterile_injections
history_of_alcohol_consumption
palpitations
scurring
inflammatory_nails
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

In [72]:
```
X_train = train_df.iloc[:, :-1].values
y_train = train_df.iloc[:, 132].values
X_test = test_df.iloc[:, :-1].values
y_test = test_df.iloc[:, 132].values
```

In [83]:
```
classifierMLP = MLPClassifier()
classifierMLP.fit(X_train, y_train)
```

```
C:\Users\damma\anaconda4\Lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:691: ConvergenceWarning: Stochastic
Optimizer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  warnings.warn(
```

Out[83]:
```
MLPClassifier
MLPClassifier()
```

In [81]:
```
classifierDT = DecisionTreeClassifier(splitter='best', criterion='entropy', min_samples_leaf=2)
classifierDT.fit(X_train, y_train)
```

Out[81]:
```
DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', min_samples_leaf=2)
```

In [82]:
```
classifierRF = RandomForestClassifier(criterion='entropy', min_samples_leaf=2)
classifierRF.fit(X_train, y_train)
```

Out[82]:
```
RandomForestClassifier
RandomForestClassifier(criterion='entropy', min_samples_leaf=2)
```

In [84]:
```
y_predMLP = classifierMLP.predict(X_test)
y_predDT = classifierDT.predict(X_test)
y_predRF = classifierRF.predict(X_test)
```

In [89]:
```
print(confusion_matrix(y_test, y_predMLP))
print(classification_report(y_test, y_predMLP))

print("Train Accuracy: ", accuracy_score(y_train, classifierMLP.predict(X_train)))
print("Test Accuracy: ", accuracy_score(y_test, y_predMLP))
```

```
[[120   0   0 ...   0   0   0]
 [  0 120   0 ...   0   0   0]
 [  0   0 120 ...   0   0   0]
 ...
 [  0   0   0 ... 120   0   0]
 [  0   0   0 ...   0 120   0]
 [  0   0   0 ...   0   0 120]]
```

	precision	recall	f1-score	support
(vertigo) Paroymsal Positional Vertigo	1.00	1.00	1.00	120
AIDS	1.00	1.00	1.00	120
Acne	1.00	1.00	1.00	120
Alcoholic hepatitis	1.00	1.00	1.00	120
Allergy	1.00	1.00	1.00	120
Arthritis	1.00	1.00	1.00	120
Bronchial Asthma	1.00	1.00	1.00	120
Cervical spondylosis	1.00	1.00	1.00	120
Chicken pox	1.00	1.00	1.00	120
Chronic cholestasis	1.00	1.00	1.00	120
Common Cold	1.00	1.00	1.00	120
Dengue	1.00	1.00	1.00	120
Diabetes	1.00	1.00	1.00	120
Dimorphic hemmorhoids(piles)	1.00	1.00	1.00	120
Drug Reaction	1.00	1.00	1.00	120
Fungal infection	1.00	1.00	1.00	120
GERD	1.00	1.00	1.00	120
Gastroenteritis	1.00	1.00	1.00	120
Heart attack	1.00	1.00	1.00	120
Hepatitis B	1.00	1.00	1.00	120
Hepatitis C	1.00	1.00	1.00	120
Hepatitis D	1.00	1.00	1.00	120
Hepatitis E	1.00	1.00	1.00	120
Hypertension	1.00	1.00	1.00	120
Hyperthyroidism	1.00	1.00	1.00	120
Hypoglycemia	1.00	1.00	1.00	120
Hypothyroidism	1.00	1.00	1.00	120
Impetigo	1.00	1.00	1.00	120
Jaundice	1.00	1.00	1.00	120
Malaria	1.00	1.00	1.00	120
Migraine	1.00	1.00	1.00	120
Osteoarthristis	1.00	1.00	1.00	120
Paralysis (brain hemorrhage)	1.00	1.00	1.00	120
Peptic ulcer diseae	1.00	1.00	1.00	120
Pneumonia	1.00	1.00	1.00	120

```
                          Psoriasis       1.00      1.00      1.00       120
                       Tuberculosis       1.00      1.00      1.00       120
                            Typhoid       1.00      1.00      1.00       120
            Urinary tract infection       1.00      1.00      1.00       120
                     Varicose veins       1.00      1.00      1.00       120
                       hepatitis A        1.00      1.00      1.00       120

                           accuracy                           1.00      4920
                          macro avg       1.00      1.00      1.00      4920
                       weighted avg       1.00      1.00      1.00      4920

Train Accuracy:  1.0
Test Accuracy:  1.0
```

In [86]:

```
print(confusion_matrix(y_test, y_predRF))
print(classification_report(y_test, y_predRF))


print("Train Accuracy: ", accuracy_score(y_train, classifierRF.predict(X_train)))
print("Test Accuracy: ", accuracy_score(y_test, y_predRF))
```

```
[[114   0   0 ...   0   0   0]
 [  0 114   0 ...   0   0   0]
 [  0   0   0 ...   0   0   0]
 ...
 [  0   0   0 ...   0   0   0]
 [  0   0   0 ...   0 120   0]
 [  0   0   0 ...   0   0 120]]
```

	precision	recall	f1-score	support
(vertigo) Paroymsal Positional Vertigo	1.00	0.95	0.97	120
AIDS	0.95	0.95	0.95	120
Acne	0.00	0.00	0.00	120
Alcoholic hepatitis	1.00	0.95	0.97	120
Allergy	0.00	0.00	0.00	120
Arthritis	1.00	1.00	1.00	120
Bronchial Asthma	1.00	1.00	1.00	120
Cervical spondylosis	0.91	1.00	0.95	120
Chicken pox	1.00	1.00	1.00	120
Chronic cholestasis	1.00	1.00	1.00	120
Common Cold	1.00	1.00	1.00	120
Dengue	1.00	1.00	1.00	120
Diabetes	1.00	1.00	1.00	120
Dimorphic hemmorhoids(piles)	0.31	1.00	0.48	120
Drug Reaction	0.00	0.00	0.00	120
Fungal infection	0.20	1.00	0.34	120
GERD	1.00	1.00	1.00	120
Gastroenteritis	0.90	0.90	0.90	120
Heart attack	1.00	1.00	1.00	120
Hepatitis B	1.00	1.00	1.00	120
Hepatitis C	1.00	1.00	1.00	120
Hepatitis D	1.00	0.05	0.10	120
Hepatitis E	0.53	1.00	0.69	120
Hypertension	0.95	0.90	0.92	120
Hyperthyroidism	1.00	1.00	1.00	120
Hypoglycemia	1.00	1.00	1.00	120
Hypothyroidism	1.00	1.00	1.00	120
Impetigo	0.00	0.00	0.00	120
Jaundice	1.00	1.00	1.00	120
Malaria	1.00	1.00	1.00	120
Migraine	0.91	1.00	0.95	120
Osteoarthristis	1.00	1.00	1.00	120
Paralysis (brain hemorrhage)	1.00	0.80	0.89	120
Peptic ulcer diseae	0.95	1.00	0.98	120
Pneumonia	1.00	1.00	1.00	120

```
                Psoriasis       0.00      0.00      0.00       120
             Tuberculosis       1.00      1.00      1.00       120
                  Typhoid       1.00      1.00      1.00       120
  Urinary tract infection       0.00      0.00      0.00       120
           Varicose veins       1.00      1.00      1.00       120
              hepatitis A       0.95      1.00      0.98       120

                 accuracy                           0.82      4920
                macro avg       0.79      0.82      0.78      4920
             weighted avg       0.79      0.82      0.78      4920

Train Accuracy:  0.8333333333333334
Test Accuracy:  0.8170731707317073
```

```
C:\Users\damma\anaconda4\Lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision and F-scor
e are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavi
or.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\damma\anaconda4\Lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision and F-scor
e are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavi
or.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\damma\anaconda4\Lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision and F-scor
e are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavi
or.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [87]:

```
print(confusion_matrix(y_test, y_predDT))
print(classification_report(y_test, y_predDT))


print("Train Accuracy: ", accuracy_score(y_train, classifierDT.predict(X_train)))
print("Test Accuracy: ", accuracy_score(y_test, y_predDT))
```

```
[[108   6   0 ...   0   0   0]
 [  0 120   0 ...   0   0   0]
 [  0   6 114 ...   0   0   0]
 ...
 [  0 120   0 ...   0   0   0]
 [  0   6   0 ...   0   0   0]
 [  0   6   0 ...   0   0   0]]
```

	precision	recall	f1-score	support
(vertigo) Paroymsal Positional Vertigo	0.33	0.90	0.49	120
AIDS	0.15	1.00	0.26	120
Acne	0.32	0.95	0.47	120
Alcoholic hepatitis	0.46	0.90	0.61	120
Allergy	0.00	0.00	0.00	120
Arthritis	0.50	0.95	0.66	120
Bronchial Asthma	0.45	0.75	0.57	120
Cervical spondylosis	0.49	0.95	0.64	120
Chicken pox	0.32	0.90	0.47	120
Chronic cholestasis	0.47	0.85	0.61	120
Common Cold	0.48	0.80	0.60	120
Dengue	0.42	0.85	0.57	120
Diabetes	0.29	0.95	0.44	120
Dimorphic hemmorhoids(piles)	0.00	0.00	0.00	120
Drug Reaction	0.00	0.00	0.00	120
Fungal infection	0.47	0.80	0.59	120
GERD	0.29	0.90	0.43	120
Gastroenteritis	0.00	0.00	0.00	120
Heart attack	0.00	0.00	0.00	120
Hepatitis B	0.34	0.95	0.50	120
Hepatitis C	0.00	0.00	0.00	120
Hepatitis D	0.00	0.00	0.00	120
Hepatitis E	0.00	0.00	0.00	120
Hypertension	0.00	0.00	0.00	120
Hyperthyroidism	0.49	0.90	0.63	120
Hypoglycemia	0.00	0.00	0.00	120
Hypothyroidism	0.00	0.00	0.00	120
Impetigo	0.00	0.00	0.00	120
Jaundice	0.00	0.00	0.00	120
Malaria	0.00	0.00	0.00	120
Migraine	0.00	0.00	0.00	120
Osteoarthristis	0.00	0.00	0.00	120
Paralysis (brain hemorrhage)	0.00	0.00	0.00	120
Peptic ulcer diseae	0.00	0.00	0.00	120
Pneumonia	0.00	0.00	0.00	120

```
               Psoriasis       0.00      0.00      0.00       120
            Tuberculosis       0.00      0.00      0.00       120
                 Typhoid       0.00      0.00      0.00       120
 Urinary tract infection       0.00      0.00      0.00       120
          Varicose veins       0.00      0.00      0.00       120
             hepatitis A       0.00      0.00      0.00       120

                accuracy                           0.35      4920
               macro avg       0.15      0.35      0.21      4920
            weighted avg       0.15      0.35      0.21      4920

Train Accuracy:  0.40476190476190477
Test Accuracy:  0.34878048780487804878
```

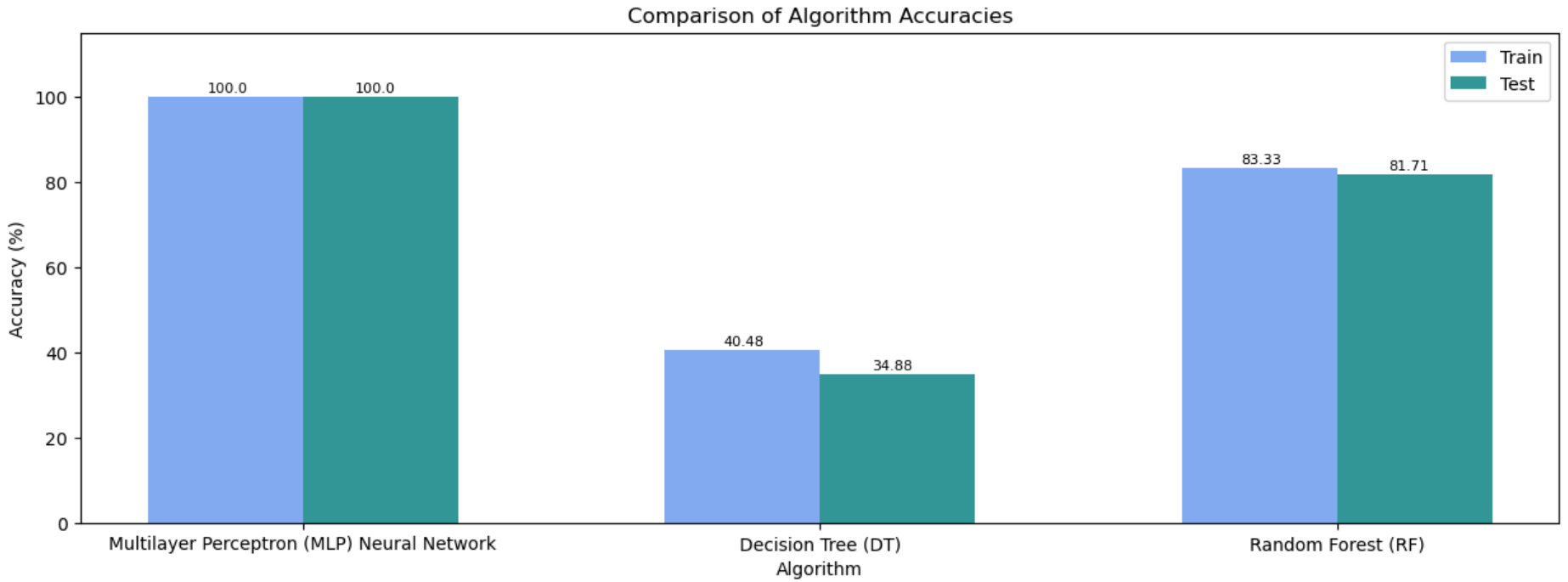
```
C:\Users\damma\anaconda4\Lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision and F-scor
e are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavi
or.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\damma\anaconda4\Lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision and F-scor
e are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavi
or.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\damma\anaconda4\Lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision and F-scor
e are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavi
or.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [88]:

```
# data to plot
n_groups = 3
algorithms = ('Multilayer Perceptron (MLP) Neural Network', 'Decision Tree (DT)', 'Random Forest (RF)')
train_accuracy = (accuracy_score(y_train, classifierMLP.predict(X_train))*100,
                  accuracy_score(y_train, classifierDT.predict(X_train))*100,
                  accuracy_score(y_train, classifierRF.predict(X_train))*100)
test_accuracy = (accuracy_score(y_test, y_predMLP)*100,
                 accuracy_score(y_test, y_predDT)*100,
                 accuracy_score(y_test, y_predRF)*100)

# create plot
fig, ax = plt.subplots(figsize=(15, 5))
index = np.arange(n_groups)
bar_width = 0.3
opacity = 0.8
rects1 = plt.bar(index, train_accuracy, bar_width, alpha = opacity, color='Cornflowerblue', label='Train')
rects2 = plt.bar(index + bar_width, test_accuracy, bar_width, alpha = opacity, color='Teal', label='Test')
```

```
plt.xlabel('Algorithm') # x axis label
plt.ylabel('Accuracy (%)') # y axis label
plt.ylim(0, 115)
plt.title('Comparison of Algorithm Accuracies') # plot title
plt.xticks(index + bar_width * 0.5, algorithms) # x axis data labels
plt.legend(loc = 'upper right') # show legend
for index, data in enumerate(train_accuracy):
    plt.text(x = index - 0.035, y = data + 1, s = round(data, 2), fontdict = dict(fontsize = 8))
for index, data in enumerate(test_accuracy):
    plt.text(x = index + 0.25, y = data + 1, s = round(data, 2), fontdict = dict(fontsize = 8))
plt.show()
```



In [ ]: