

★

# Classification Model to Identify Multiple Disease

In [1]:

```
# import library
import pandas as pd
```

In [3]:

```
# import data
disease = pd.read_csv('MultipleDiseasePrediction.csv')
```

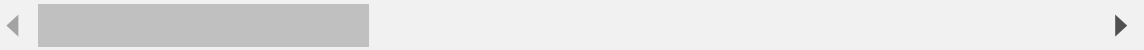
In [4]:

```
# view data
disease.head()
```

Out[4]:

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

5 rows × 133 columns



In [7]:

```
# info of data
disease.info(verbose= True, show_counts= True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4920 entries, 0 to 4919
Data columns (total 133 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   itching                               4920 non-null   int64
1   skin_rash                             4920 non-null   int64
2   nodal_skin_eruptions                  4920 non-null   int64
3   continuous_sneezing                   4920 non-null   int64
4   shivering                              4920 non-null   int64
5   chills                                4920 non-null   int64
6   joint_pain                            4920 non-null   int64
7   stomach_pain                          4920 non-null   int64
8   acidity                               4920 non-null   int64
9   ulcers_on_tongue                      4920 non-null   int64
10  muscle_wasting                         4920 non-null   int64
11  vomiting                               4920 non-null   int64
12  burning_micturition                    4920 non-null   int64
13  spotting_urination                     4920 non-null   int64
..  ..
```

In [5]:

```
# summary statistics
disease.describe()
```

Out[5]:

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	
count	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4
mean	0.137805	0.159756	0.021951	0.045122	0.021951	
std	0.344730	0.366417	0.146539	0.207593	0.146539	
min	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	0.000000	0.000000	
75%	0.000000	0.000000	0.000000	0.000000	0.000000	
max	1.000000	1.000000	1.000000	1.000000	1.000000	

8 rows × 132 columns

In [8]:

```
# check for missing value
disease.isnull().sum()
```

Out[8]:

```
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 [7]:

```
# check for categories
disease.nunique()
```

Out[7]:

```
itching                2
skin_rash              2
nodal_skin_eruptions   2
continuous_sneezing    2
shivering              2
..
inflammatory_nails     2
blister                2
red_sore_around_nose   2
yellow_crust_ooze      2
prognosis              41
Length: 133, dtype: int64
```

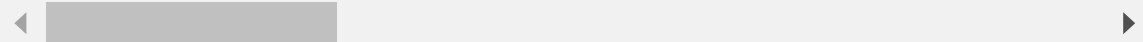
In [8]:

```
# correlation
disease.corr()
```

Out[8]:

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering
itching	1.000000	0.318158	0.326439	-0.086906	-0.059893
skin_rash	0.318158	1.000000	0.298143	-0.094786	-0.065324
nodal_skin_eruptions	0.326439	0.298143	1.000000	-0.032566	-0.022444
continuous_sneezing	-0.086906	-0.094786	-0.032566	1.000000	0.608981
shivering	-0.059893	-0.065324	-0.022444	0.608981	1.000000
...	...	...	...	...	...
small_dents_in_nails	-0.061573	0.331087	-0.023073	-0.033480	-0.033480
inflammatory_nails	-0.061573	0.331087	-0.023073	-0.033480	-0.033480
blister	-0.061573	0.331087	-0.023073	-0.033480	-0.033480
red_sore_around_nose	-0.061573	0.331087	-0.023073	-0.033480	-0.033480
yellow_crust_ooze	-0.061573	0.331087	-0.023073	-0.033480	-0.033480

132 rows × 132 columns



In [9]:

```
# visualize pairplot

import seaborn as sns
```

In [ ]:

```
# There is no logic of using pairplot as it has 133 columns.
# sns.pairplot(disease)
```

In [11]:

```
# column names
disease.columns
```

Out[11]:

```
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 [12]:

```
# define y
y=disease['prognosis']
```

In [13]:

```
# define X
X=disease.drop(['prognosis'], axis=1)
```

In [14]:

```
# import function
from sklearn.model_selection import train_test_split
```

In [15]:

```
# split data
X_train, X_test, y_train, y_test = train_test_split(X,y, train_size=0.8, random_state=25)
```

In [16]:

```
# verify shape
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

Out[16]:

```
((3936, 132), (984, 132), (3936,), (984,))
```

In [17]:

```
# select model
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
```

In [18]:

```
# train model
model.fit(X_train,y_train)
```

Out[18]:

```
RandomForestClassifier()
```

In [19]:

```
# predict with model
y_pred=model.predict(X_test)
```

In [20]:

```
# model evaluation
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

In [21]:

```
# model accuracy
accuracy_score(y_test, y_pred)
```

Out[21]:

1.0

In [22]:

```
# model confusion matrix
confusion_matrix(y_test, y_pred)
```

Out[22]:

```
array([[20,  0,  0, ...,  0,  0,  0],
       [ 0, 22,  0, ...,  0,  0,  0],
       [ 0,  0, 22, ...,  0,  0,  0],
       ...,
       [ 0,  0,  0, ..., 27,  0,  0],
       [ 0,  0,  0, ...,  0, 23,  0],
       [ 0,  0,  0, ...,  0,  0, 23]], dtype=int64)
```

In [23]:

```
# model classification report  
print(classification_report(y_test,y_pred))
```

support		precision	recall	f1-score
(vertigo) Paroymsal	Positional Vertigo	1.00	1.00	1.00
20				
	AIDS	1.00	1.00	1.00
22				
	Acne	1.00	1.00	1.00
22				
	Alcoholic hepatitis	1.00	1.00	1.00
28				
	Allergy	1.00	1.00	1.00
21				
	Arthritis	1.00	1.00	1.00
28				
	Bronchial Asthma	1.00	1.00	1.00
25				
	Cervical spondylosis	1.00	1.00	1.00
20				
	Chicken pox	1.00	1.00	1.00
23				
	Chronic cholestasis	1.00	1.00	1.00
23				
	Common Cold	1.00	1.00	1.00
27				
	Dengue	1.00	1.00	1.00
20				
	Diabetes	1.00	1.00	1.00
23				
	Dimorphic hemmorhoids(piles)	1.00	1.00	1.00
23				
	Drug Reaction	1.00	1.00	1.00
20				
	Fungal infection	1.00	1.00	1.00
24				
	GERD	1.00	1.00	1.00
17				
	Gastroenteritis	1.00	1.00	1.00
26				
	Heart attack	1.00	1.00	1.00
26				
	Hepatitis B	1.00	1.00	1.00
29				
	Hepatitis C	1.00	1.00	1.00
22				
	Hepatitis D	1.00	1.00	1.00
27				
	Hepatitis E	1.00	1.00	1.00
18				
	Hypertension	1.00	1.00	1.00
23				
	Hyperthyroidism	1.00	1.00	1.00
24				
	Hypoglycemia	1.00	1.00	1.00
22				
	Hypothyroidism	1.00	1.00	1.00
19				
	Impetigo	1.00	1.00	1.00
33				
	Jaundice	1.00	1.00	1.00
26				



	Malaria	1.00	1.00	1.00
28				
	Migraine	1.00	1.00	1.00
22				
	Osteoarthritis	1.00	1.00	1.00
30				
	Paralysis (brain hemorrhage)	1.00	1.00	1.00
28				
	Peptic ulcer disease	1.00	1.00	1.00
23				
	Pneumonia	1.00	1.00	1.00
21				
	Psoriasis	1.00	1.00	1.00
24				
	Tuberculosis	1.00	1.00	1.00
30				
	Typhoid	1.00	1.00	1.00
24				
	Urinary tract infection	1.00	1.00	1.00
27				
	Varicose veins	1.00	1.00	1.00
23				
	hepatitis A	1.00	1.00	1.00
23				
	accuracy			1.00
984				
	macro avg	1.00	1.00	1.00
984				
	weighted avg	1.00	1.00	1.00
984				

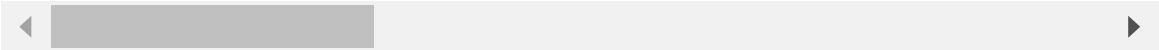
In [24]:

```
# future prediction
sample=disease.sample()
sample
```

Out[24]:

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_
4747	0	0	0	0	0	0	

1 rows × 133 columns



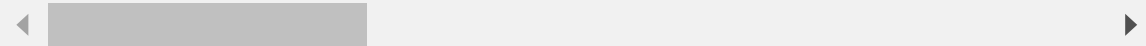
In [25]:

```
# define X_new
X_new = X.sample()
X_new.sample()
```

Out[25]:

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_
4771	0	0	0	0	0	1	

1 rows × 132 columns



In [26]:

```
# predict for X_new
model.predict(X_new)
```

Out[26]:

```
array(['Malaria'], dtype=object)
```

In [ ]: