

▼

★

Classification Model to Identify Multiple Disease

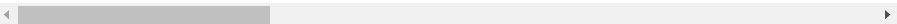
```
# import library
import pandas as pd

# import data
disease = pd.read_csv('https://github.com/ybifoundation/Dataset/raw/main/MultipleDiseasePrediction.csv')

# view data
disease.head()
```

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joi
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



```
# info of data
disease.info()

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

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

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shive
count	4920.000000	4920.000000	4920.000000	4920.000000	4920.00
mean	0.137805	0.159756	0.021951	0.045122	0.02
std	0.344730	0.366417	0.146539	0.207593	0.14
min	0.000000	0.000000	0.000000	0.000000	0.00
25%	0.000000	0.000000	0.000000	0.000000	0.00
50%	0.000000	0.000000	0.000000	0.000000	0.00
75%	0.000000	0.000000	0.000000	0.000000	0.00
max	1.000000	1.000000	1.000000	1.000000	1.00

8 rows × 132 columns



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

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

```

```

# check for categories
disease.nunique()

```

```

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

```

```

# correlation
disease.corr()

```

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	small_dents_in_nails	inflammatory_nails	blister	red_sore_around_nose	yellow_crust_ooze
itching	1.000000	0.318158	0.326439	-0.086906	-0.059893	-0.061573	-0.061573	-0.061573	-0.061573	-0.061573	-0.061573
skin_rash	0.318158	1.000000	0.298143	-0.094786	-0.065324	0.331087	0.331087	0.331087	0.331087	0.331087	0.331087
nodal_skin_eruptions	0.326439	0.298143	1.000000	-0.032566	-0.022444	-0.023073	-0.023073	-0.023073	-0.023073	-0.023073	-0.023073
continuous_sneezing	-0.086906	-0.094786	-0.032566	1.000000	0.608197	-0.033073	-0.033073	-0.033073	-0.033073	-0.033073	-0.033073
shivering	-0.059893	-0.065324	-0.022444	0.608197	1.000000	-0.033073	-0.033073	-0.033073	-0.033073	-0.033073	-0.033073
chills	-0.061573	0.331087	-0.023073	-0.033073	-0.033073	1.000000	0.331087	0.331087	0.331087	0.331087	0.331087
small_dents_in_nails	-0.061573	0.331087	-0.023073	-0.033073	-0.033073	0.331087	1.000000	1.000000	1.000000	1.000000	1.000000
inflammatory_nails	-0.061573	0.331087	-0.023073	-0.033073	-0.033073	0.331087	1.000000	1.000000	1.000000	1.000000	1.000000
blister	-0.061573	0.331087	-0.023073	-0.033073	-0.033073	0.331087	1.000000	1.000000	1.000000	1.000000	1.000000
red_sore_around_nose	-0.061573	0.331087	-0.023073	-0.033073	-0.033073	0.331087	1.000000	1.000000	1.000000	1.000000	1.000000
yellow_crust_ooze	-0.061573	0.331087	-0.023073	-0.033073	-0.033073	0.331087	1.000000	1.000000	1.000000	1.000000	1.000000

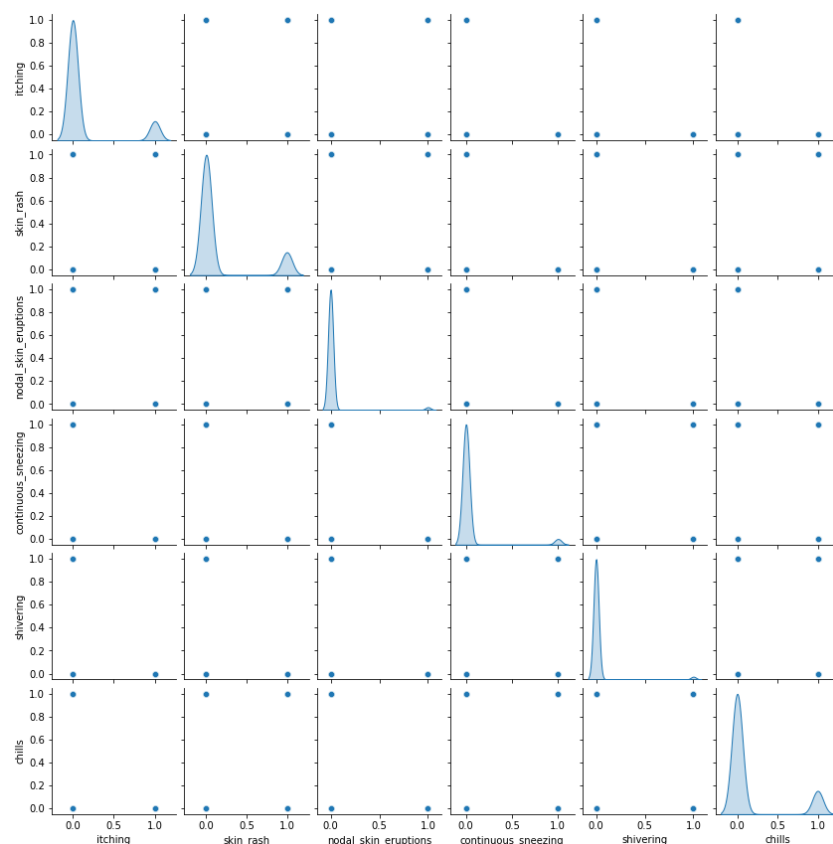
132 rows × 132 columns



```

# visualize pairplot
from IPython.display import Image
import seaborn as sns
import matplotlib.pyplot as plt
df = disease[['itching', 'skin_rash', 'nodal_skin_eruptions', 'continuous_sneezing',
'shivering', 'chills']]
sns_plot = sns.pairplot(df,diag_kind='kde',height=2.0)
sns_plot.savefig("pairplot.png")
plt.clf()
Image(filename='pairplot.png')

```



<Figure size 864x864 with 0 Axes>

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

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

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

```
# define X
X = disease[['itching', 'skin_rash', 'nodal_skin_eruptions', 'continuous_sneezing', 'shivering', 'chills', 'joint_pain', 'stomach_pain', 'acidity', 'u
```

```
X.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4920 entries, 0 to 4919
Columns: 132 entries, itching to yellow_crust_ooze
dtypes: int64(132)
memory usage: 5.0 MB
```

```
# split data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 0.7, random_state=2529 )
```

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

```
((3444, 132), (1476, 132), (3444,), (1476,))
```

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

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

```
RandomForestClassifier()
```

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

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

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

```
1.0
```

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

```
array([[31,  0,  0, ...,  0,  0,  0],
       [ 0, 37,  0, ...,  0,  0,  0],
       [ 0,  0, 35, ...,  0,  0,  0],
       ...,
       [ 0,  0,  0, ..., 39,  0,  0],
       [ 0,  0,  0, ...,  0, 35,  0],
       [ 0,  0,  0, ...,  0,  0, 32]])
```

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

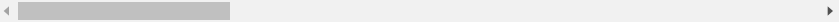
	precision	recall	f1-score	support
(vertigo) Paroymsal	1.00	1.00	1.00	31
AIDS	1.00	1.00	1.00	37
Acne	1.00	1.00	1.00	35
Alcoholic hepatitis	1.00	1.00	1.00	40
Allergy	1.00	1.00	1.00	37
Arthritis	1.00	1.00	1.00	46
Bronchial Asthma	1.00	1.00	1.00	37
Cervical spondylosis	1.00	1.00	1.00	31
Chicken pox	1.00	1.00	1.00	29
Chronic cholestasis	1.00	1.00	1.00	32
Common Cold	1.00	1.00	1.00	39
Dengue	1.00	1.00	1.00	35
Diabetes	1.00	1.00	1.00	35
Dimorphic hemmorhoids(piles)	1.00	1.00	1.00	34
Drug Reaction	1.00	1.00	1.00	38
Fungal infection	1.00	1.00	1.00	35
GERD	1.00	1.00	1.00	31
Gastroenteritis	1.00	1.00	1.00	36
Heart attack	1.00	1.00	1.00	41
Hepatitis B	1.00	1.00	1.00	46
Hepatitis C	1.00	1.00	1.00	32
Hepatitis D	1.00	1.00	1.00	39
Hepatitis E	1.00	1.00	1.00	29
Hypertension	1.00	1.00	1.00	33
Hyperthyroidism	1.00	1.00	1.00	36
Hypoglycemia	1.00	1.00	1.00	33
Hypothyroidism	1.00	1.00	1.00	30
Impetigo	1.00	1.00	1.00	48
Jaundice	1.00	1.00	1.00	36
Malaria	1.00	1.00	1.00	41
Migraine	1.00	1.00	1.00	38
Osteoarthritis	1.00	1.00	1.00	38
Paralysis (brain hemorrhage)	1.00	1.00	1.00	42
Peptic ulcer diseae	1.00	1.00	1.00	29
Pneumonia	1.00	1.00	1.00	33
Psoriasis	1.00	1.00	1.00	33
Tuberculosis	1.00	1.00	1.00	42
Typhoid	1.00	1.00	1.00	33
Urinary tract infection	1.00	1.00	1.00	39
Varicose veins	1.00	1.00	1.00	35
hepatitis A	1.00	1.00	1.00	32

accuracy			1.00	1476
macro avg	1.00	1.00	1.00	1476
weighted avg	1.00	1.00	1.00	1476

```
# future prediction
X.sample()
```

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	ch
1134	0	0	0	0	0	

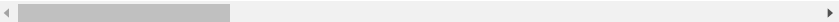
1 rows × 132 columns



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

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	ch
3371	0	0	0	0	0	

1 rows × 132 columns



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

array(['Bronchial Asthma'], dtype=object)
```