Problem Statement

The objective of this case study is to analyze the relationship between symptoms and diseases in order to accurately diagnose a person's condition. With various symptoms such as headache, joint pain, and cough, it is crucial to determine the underlying disease, as it can range from mild to severe. This study highlights the significance of identifying specific symptoms and their corresponding diseases to improve diagnostic accuracy and facilitate appropriate medical treatment

```
import numpy as np
import pandas as pd
```

```
df=pd.read_csv('/content/dataset.csv')
df
```

	Disease	Symptom_1	Symptom_2	Symptom_3	Symptom_4	Symptom_5	Symptom_6	Sympt
0	Fungal infection	itching	skin_rash	nodal_skin_eruptions	dischromic _patches	NaN	NaN	
1	Fungal infection	skin_rash	nodal_skin_eruptions	dischromic _patches	NaN	NaN	NaN	
2	Fungal infection	itching	nodal_skin_eruptions	dischromic _patches	NaN	NaN	NaN	
3	Fungal infection	itching	skin_rash	dischromic _patches	NaN	NaN	NaN	
4	Fungal infection	itching	skin_rash	nodal_skin_eruptions	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	
4915	(vertigo) Paroymsal Positional Vertigo	vomiting	headache	nausea	spinning_movements	loss_of_balance	unsteadiness	
4916	Acne	skin_rash	pus_filled_pimples	blackheads	scurring	NaN	NaN	
4917	Urinary tract infection	burning_micturition	bladder_discomfort	foul_smell_of urine	continuous_feel_of_urine	NaN	NaN	
4918	Psoriasis	skin_rash	joint_pain	skin_peeling	silver_like_dusting	small_dents_in_nails	inflammatory_nails	
4919	Impetigo	skin_rash	high_fever	blister	red_sore_around_nose	yellow_crust_ooze	NaN	

4920 rows × 18 columns

```
df.isnull().mean()*100
```

	0
Disease	0.000000
Symptom_1	0.000000
Symptom_2	0.000000
Symptom_3	0.000000
Symptom_4	7.073171
Symptom_5	24.512195
Symptom_6	40.365854
Symptom_7	53.902439
Symptom_8	60.487805
Symptom_9	65.609756
Symptom_10	69.268293
Symptom_11	75.731707
Symptom_12	84.878049
Symptom_13	89.756098
Symptom_14	93.780488
Symptom_15	95.121951
Symptom_16	96.097561
Symptom_17	98.536585

dtype: float64

```
df.isnull().sum()
```

	0
Disease	0
Symptom_1	0
Symptom_2	0
Symptom_3	0
Symptom_4	348
Symptom_5	1206
Symptom_6	1986
Symptom_7	2652
Symptom_8	2976
Symptom_9	3228
Symptom_10	3408
Symptom_11	3726
Symptom_12	4176
Symptom_13	4416
Symptom_14	4614
Symptom_15	4680
Symptom_16	4728
Symptom_17	4848

dtype: int64

```
df_new=df.iloc[:,:7]
```

```
df_new
```

	Disease	Symptom_1	Symptom_2	Symptom_3	Symptom_4	Symptom_5	Symptom_6
0	Fungal infection	itching	skin_rash	nodal_skin_eruptions	dischromic _patches	NaN	NaN
1	Fungal infection	skin_rash	nodal_skin_eruptions	dischromic _patches	NaN	NaN	NaN
2	Fungal infection	itching	nodal_skin_eruptions	dischromic _patches	NaN	NaN	NaN
3	Fungal infection	itching	skin_rash	dischromic _patches	NaN	NaN	NaN
4	Fungal infection	itching	skin_rash	nodal_skin_eruptions	NaN	NaN	NaN
...	...	...	...	...	...	...	...
4915	(vertigo) Paroymsal Positional Vertigo	vomiting	headache	nausea	spinning_movements	loss_of_balance	unsteadiness
4916	Acne	skin_rash	pus_filled_pimples	blackheads	scurring	NaN	NaN
4917	Urinary tract infection	burning_micturition	bladder_discomfort	foul_smell_of urine	continuous_feel_of_urine	NaN	NaN
4918	Psoriasis	skin_rash	joint_pain	skin_peeling	silver_like_dusting	small_dents_in_nails	inflammatory_nails

```
for col in df_new.columns:
  df_new[col].fillna(df[col].mode()[0],inplace=True)
```

```
put-7-1497b74a2304>:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an
r will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always b

, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)


l].fillna(df[col].mode()[0],inplace=True)
put-7-1497b74a2304>:2: SettingWithCopyWarning:
trying to be set on a copy of a slice from a DataFrame

eats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
l].fillna(df[col].mode()[0],inplace=True)
```

```
df_new.isnull().sum()
```

	0
Disease	0
Symptom_1	0
Symptom_2	0
Symptom_3	0
Symptom_4	0
Symptom_5	0
Symptom_6	0

dtype: int64

```
# df_new.duplicated().sum()
```

```
np.int64(4658)
```

```
# df_new.shape
```

```
(4920, 7)
```

```
from sklearn.preprocessing import LabelEncoder
# for col in df_new.columns:
```

```
LE=LabelEncoder()
```

```
for col in df_new.columns:
  df_new[col]=LE.fit_transform(df_new[col])
```

```
<ipython-input-15-c8178ad355e2>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  df_new[col]=LE.fit_transform(df_new[col])
<ipython-input-15-c8178ad355e2>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  df_new[col]=LE.fit_transform(df_new[col])
<ipython-input-15-c8178ad355e2>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  df_new[col]=LE.fit_transform(df_new[col])
<ipython-input-15-c8178ad355e2>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  df_new[col]=LE.fit_transform(df_new[col])
<ipython-input-15-c8178ad355e2>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  df_new[col]=LE.fit_transform(df_new[col])
<ipython-input-15-c8178ad355e2>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  df_new[col]=LE.fit_transform(df_new[col])
<ipython-input-15-c8178ad355e2>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  df_new[col]=LE.fit_transform(df_new[col])
```

```
df_new
```

	Disease	Symptom_1	Symptom_2	Symptom_3	Symptom_4	Symptom_5	Symptom_6
0	15	33	35	35	11	12	20
1	15	24	27	18	19	12	20
2	15	33	27	18	19	12	20
3	15	33	35	18	19	12	20
4	15	33	35	35	19	12	20
...	...	...	...	...	...	...	...
4915	0	28	17	33	38	22	29
4916	2	24	31	3	35	12	20
4917	38	4	4	22	7	12	20
4918	35	24	20	42	36	27	15
4919	27	24	18	4	33	35	20

4920 rows × 7 columns

```
from sklearn.model_selection import train_test_split
```

```
x=df_new.drop('Disease',axis=1)
y=df_new['Disease']
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.80,random_state=42)
```

```
x_train
```

	Symptom_1	Symptom_2	Symptom_3	Symptom_4	Symptom_5	Symptom_6
1807	33	35	21	24	13	12
184	6	42	21	19	12	20
205	33	15	28	9	21	0
4581	11	12	8	30	33	28
410	33	35	35	11	12	20
...	...	...	...	...	...	...
4426	24	20	42	36	27	15
466	28	19	29	0	16	20
3092	24	9	26	44	11	13
3772	33	35	35	11	12	20
860	33	35	44	5	28	20

3936 rows × 6 columns

```
y_train
```

	Disease
1807	8
184	37
205	19
4581	39
410	15
...	...
4426	35
466	33
3092	11
3772	15
860	14

3936 rows × 1 columns

dtype: int64

```
from sklearn.neighbors import KNeighborsClassifier
```

```
KN=KNeighborsClassifier()
```

```
KN.fit(x_train,y_train)
```

```
▾ KNeighborsClassifier ⓘ ?
KNeighborsClassifier()
```

```
acc=KN.score(x_test,y_test)
```

```
acc*100
```

```
98.47560975609755
```

```
acc=KN.score(x_train,y_train)
acc*100
```

```
99.6189024390244
```

```
y_pred=KN.predict(x_test)
```

```
from sklearn.metrics import *
```

```
accuracy_score(y_test,y_pred)*100
```

```
98.47560975609755
```