

Name: Priyanka Kumari

Registration Number: 20BCE2334

DECISION TREE CLASSIFIER TO PREDICT COVID -19

ML ASSIGNMENT 2

```
In [66]: import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn import metrics
covid=pd.read_csv("Cleaned-Data.csv")
print(covid.head(2))

   Fever  Tiredness  Dry-Cough  Difficulty-in-Breathing  Sore-Throat  \
0      1          1          1              1              1          1
1      1          1          1              1              1          1

   None_Sympton  Pains  Nasal-Congestion  Runny-Nose  Diarrhea  ...  \
0              0      1              1              1          1  ...
1              0      1              1              1          1  ...

   Gender_Male  Gender_Transgender  Severity_Mild  Severity_Moderate  \
0              1                  0              1                  0
1              1                  0              1                  0

   Severity_None  Severity_Severe  Contact_Dont-Know  Contact_No  Contact_Yes  \
0              0              0              0              0              1
1              0              0              0              1              0

   Country
0      China
1      China

[2 rows x 27 columns]
```

defining X and Y

```
In [21]: #defining X and Y
feature_cols=['Fever','Tiredness','Dry-Cough','Sore-Throat','Difficulty-in-Breathing','Pains','Nasal-Congestion','Runny-Nose','Diarrhea']
#As X is a matrix so we use [] to access the features we want in feature_cols
X=covid[feature_cols]
print(X)

   Fever  Tiredness  Dry-Cough  Difficulty-in-Breathing  Pains  \
0      1          1          1              1              1
1      1          1          1              1              1
2      1          1          1              1              1
3      1          1          1              1              1
4      1          1          1              1              1
...      ...      ...      ...      ...      ...
316795    0          0          0              0              0
316796    0          0          0              0              0
316797    0          0          0              0              0
316798    0          0          0              0              0
316799    0          0          0              0              0

   Nasal-Congestion  Runny-Nose  Diarrhea
0                  1          1          1
1                  1          1          1
2                  1          1          1
3                  1          1          1
4                  1          1          1
...      ...      ...      ...
316795            0          0          0
316796            0          0          0
316797            0          0          0
316798            0          0          0
316799            0          0          0

[316800 rows x 8 columns]
```

```
In [23]: #y is a vector ,so we can use dot to access 'Severity_Severe'
Y=covid.Severity_Severe
print(Y)

0      0
1      0
2      0
3      0
4      0
..
316795    1
316796    1
316797    0
316798    0
316799    0
Name: Severity_Severe, Length: 316800, dtype: int64
```

Splitting X and Y into training and testing data

```
In [24]: X_train,X_test,Y_train,Y_test=train_test_split(X,Y,random_state=0,train_size=0.8)
```

Importing Decision Tree Classifier from sklearn

```
In [25]: from sklearn.tree import DecisionTreeClassifier
```

```
In [26]: clf=DecisionTreeClassifier(criterion='entropy')
```

Training Decision Tree Classifier

```
In [27]: clf.fit(X_train,Y_train)
```

```
Out[27]: DecisionTreeClassifier(criterion='entropy')
```

```
In [28]: #predicting the labels on test set
y_pred=clf.predict(X_test)
```

Calculating Accuracy

```
In [48]: from sklearn import metrics
print('Accuracy Score on test data is:',metrics.accuracy_score(Y_test,y_pred)*100)

Accuracy Score on test data is: 74.96527777777777
```

```
In [49]: print(metrics.confusion_matrix(Y_test,y_pred))

[[47498    0]
 [15862    0]]
```

```
In [43]: #save confusion matrix and slice into four pieces
confusion=metrics.confusion_matrix(Y_test,y_pred)
print(confusion)
TP=confusion[1,1]
TN=confusion[0,0]
FP=confusion[0,1]
FN=confusion[1,0]

[[47498    0]
 [15862    0]]
```

```
In [62]: #use float to perform true division,not integer division
#from sklearn import metrics
from sklearn.metrics import accuracy_score
print((TP+TN)/float(TP+TN+FP+FN))
print('Accuracy Score on confusion_matrix is:',accuracy_score(Y_test,y_pred,normalize=True,sample_weight=None))

0.7496527777777777
Accuracy Score on confusion_matrix is: 0.7496527777777777
```

```
In [61]: classification_error=(FP+FN)/float(TP+TN+FP+FN)
print(classification_error)
print('Error Score on confusion matrix:',1-metrics.accuracy_score(Y_test,y_pred))

0.2503472222222222
Error Score on confusion matrix: 0.2503472222222223
```

```
In [63]: sensitivity=TP/float(FN+TP)
print(sensitivity)
print('sensitivity Score on confusion matrix:',metrics.recall_score(Y_test,y_pred))

0.0
sensitivity Score on confusion matrix: 0.0
```

```
In [64]: specificity=TN/(TN+FP)
print(specificity)
print('specificity Score on confusion_matrix:',specificity)

1.0
specificity Score on confusion_matrix: 1.0
```

```
In [65]: precision=TP/float(TP+FP)
print(precision)
print('precision Score on confusion matrix:',metrics.precision_score(Y_test,y_pred))

nan
precision Score on confusion matrix: 0.0
<ipython-input-65-78085f27110a>:1: RuntimeWarning: invalid value encountered in true_divide
precision=TP/float(TP+FP)
C:\Users\Priyanka Kumari\anaconda3\lib\site-packages\sklearn\metrics\classification.py:1245: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 due to no predicted samples. Use `zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```

```
In [ ]:
```