# CSE 4020 - MACHINE LEARNING

# Lab 29+30

# Lab Task1

# Submitted by: Alokam Nikhitha(19BCE2555)

**Question:**

Build a classifier using decision tree to predict the

COVID-19 severity.

**Dataset Used:**

<https://www.kaggle.com/hemanthhari/symptomsand-covid-presence>

Procedure:

1. Import the dataset using pandas.
2. Specifying if the attributes to be used as independent attributes or dependent ones
3. We have to change categorical values to numbers
4. Split our Dataset into training set
5. Create an instance of our decision tree classifier.
6. We fit our training sets to the object of decision tree classifier in order to train it.
7. Result is stored in another array
8. We create the Confusion Matrix and check performance.

Code:

**# Importing libraries**

import numpy as np

import pandas as pd

**# Importing dataset**

data = pd.read\_csv("Covid Dataset.csv")

X = data.iloc[2500:5500, 0:6].values

y = data.iloc[2500:5500, 20:].values

**# Encoding Categorical Attribute**

from sklearn.preprocessing import OneHotEncoder, LabelEncoder

X[:, 0] = LabelEncoder().fit\_transform(X[:, 0])

X[:, 1] = LabelEncoder().fit\_transform(X[:, 1])

X[:, 2] = LabelEncoder().fit\_transform(X[:, 2])

X[:, 3] = LabelEncoder().fit\_transform(X[:, 3])

X[:, 4] = LabelEncoder().fit\_transform(X[:, 4])

X[:, 5] = LabelEncoder().fit\_transform(X[:, 5])

y[:, 0] = LabelEncoder().fit\_transform(y[:, 0])

y=y.astype('int')

**# Splitting the dataset into the training set and test set**

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=0)

**# Fitting Decision Tree Classification model to the training set**

from sklearn.tree import DecisionTreeClassifier

classifier = DecisionTreeClassifier(criterion = 'entropy', random\_state=5)

classifier.fit(X\_train, y\_train)

**# Predicting result for test set**

y\_pred = classifier.predict(X\_test)

**# Accuracy of our model**

from sklearn.metrics import accuracy\_score

accuracy\_score(y\_test, y\_pred, normalize=True, sample\_weight=None)

**#Classification error**

1-accuracy\_score(y\_test, y\_pred, normalize=True, sample\_weight=None)

**# Sensitivity**

from sklearn.metrics import recall\_score

recall\_score(y\_test, y\_pred)

**# Precision**

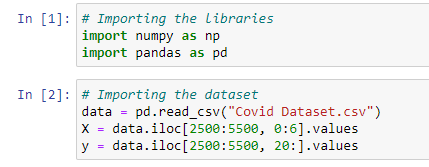
from sklearn.metrics import precision\_score

precision\_score(y\_test, y\_pred)

**#confusion Matrix**

from sklearn.metrics import confusion\_matrix

confusion\_matrix(y\_test, y\_pred)



In Cell 1:

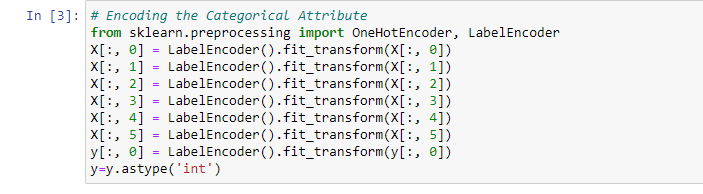
Firstly, we import the libraries numpy as np ,pandas as pd.

In Cell 2:

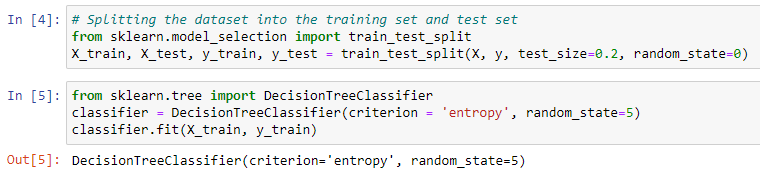
Dataset is imported from Covid Dataset.csv

We take only first 6 attributes to train our model,

The attributes are Breathing Problem, Fever, Dry Cough, Sore Throat, Running Nose and Asthma. The last column is our label attribute and it tells if a person with given symptoms had COVID-19 or not.

 In Cell 3:

We need to Encode the data as it contains yes or No, So we need to encode it into numbers



In Cell 4:

Here we have split our dataset into training set and test set. The training set will be used to train our decision tree classifier and the test will help us analyse the efficacy of trained model.

In Cell 5:

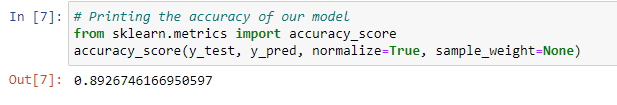
Here we have trained our decision tree classifier with training dataset.

Also, we have set the criterion to entropy and thus the decision tree that we constructed will be based on information gain

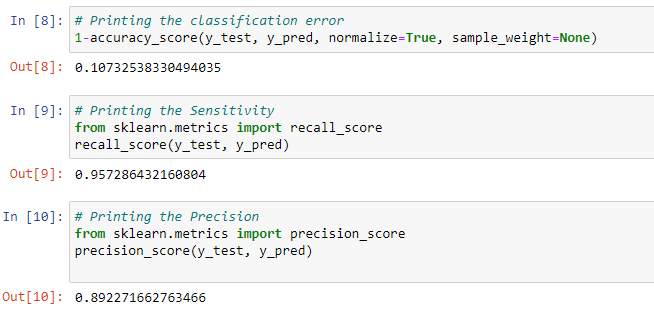


In Cell 6:

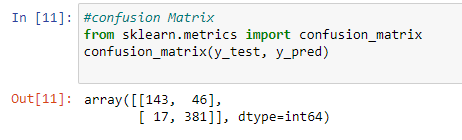
Here we are creating an array and storing the results of X\_test dataset as predicted by our classifier.



In Cell 7: Accuracy of the Model



In Cell 8,9,10: Printing Classification error,Senstivity,Precision



In cell 11:Printing Confusion Matrix

Caluculations:

Our confusion matrix is:

|  |  |
| --- | --- |
| TN  143 | FP  46 |
| FN  17 | **TP**  **381** |

# Accuracy of Model = (TP + TN) / (TP + TN + FP + FN)

=(381+143)/(381+143+46+17)

=0.8926746167(As in Cell 7)

# Classification Error = 1 – Accuracy

=1-0.8926746167

=0.1073253833(As in Cell 8)

# Sensitivity = TP / (TP + FN)

=381/(381+17)

=0.95728643216(As in Cell 9)

# Specificity = TN / (TN + FP)

=143/(143+46)

=0.75661375661

# Precision = TP / (TP + FP)

=381/(381+46)

=0.89227166276(As in Cell 10)

**Final results:**

**Accuracy =0.8926746167**

**Classification Error = 0.1073253833**

**Sensitivity = 0.95728643216**

**Specificity = 0.75661375661**

**Precision = 0.89227166276**