```
try:
  from google.colab import drive
  %tensorflow_version 2.x
  COLAB = True
  print("Note: using Google CoLab")
except:
  print("Note: not using Google CoLab")
  COLAB = False

# Print your name and Roll No.
print('Name: Rohit Byas Sherwan')
print('Roll No. :181210043')

# Print the curent time
from datetime import datetime
print("Date Time :", datetime.now())
```

```
     Note: using Google CoLab
     Name: Rohit Byas Sherwan
     Roll No. :181210043
     Date Time : 2021-03-08 08:26:32.689899
```

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

```
# creating the play tennis dataset
data_table = {'outlook':['sunny', 'sunny', 'overcast', 'rainy', 'rainy', 'rainy', 'overcast',
        'temperature':['hot', 'hot', 'hot', 'mild', 'cool', 'cool', 'cool', 'mild', 'cool', '
        'humidity':['high', 'high', 'high', 'high', 'normal', 'normal', 'normal', 'high', 'no
        'windy':['false', 'true', 'false', 'false', 'false', 'true', 'true', 'false', 'false'
        'play tennis':['no', 'no', 'yes', 'yes', 'yes', 'no', 'yes', 'no', 'yes', 'yes', 'yes

# converting the dataset into a dataframe
play_tennis_dataset = pd.DataFrame(data = data_table)
```

```
play_tennis_dataset
```

	outlook	temperature	humidity	windy	play tennis
0	sunny	hot	high	false	no
1	sunny	hot	high	true	no
2	overcast	hot	high	false	yes
3	rainy	mild	high	false	yes
4	rainy	cool	normal	false	yes
5	rainy	cool	normal	true	no
6	overcast	cool	normal	true	yes
7	sunny	mild	high	false	no
9	rainy	mild	normal	false	yes

Task 1: Use the Gini as impurity paramter and construct a Decision Tree

```
# function to compute gini impurity or gini index
def gini(dataset, class_label, feature):

    # initialize the dictionary to store gini-impurity vaue for each category of the selected
    gini_imp = {feature:{}}

    # iterating through each category of the feature
    for cat in dataset[feature].unique():

        # capture the number of datapoints belonging to a particular category
        n = dataset[dataset[feature] == cat].shape[0]

        # number of datapoints belonging to each class label of a particular category
        data_distb = dataset[dataset[feature] == cat].groupby(by = class_label)[class_label].

        # calculating squared probability of each class label in a particular category and th
        sq_prob = sum([(i/n)**2 for i in data_distb])

        # subtracting the sum of squared probabilities from one to complete the gini-impurity
        gini_impurity = np.round(1 - sq_prob, 2)

        # adding gini-impurity value for each category of the feature
        gini_imp[feature].update({cat:gini_impurity})
        #gini_imp[feature][cat] = gini_impurity # this way also you can add each category of

    # returning the dictionary for gini-impurity value for each category of the feature
    return gini_imp
```

```
# calling the function to calculate gini-impurity of each category of a feature
gini(play_tennis_dataset, 'play tennis', 'humidity')
```

```
{'humidity': {'high': 0.49, 'normal': 0.24}}
```