## Introduction

In this project, the objective is to write python code for creating “ID3 Decision Tree” classifier from scratch.

## About Dataset

The dataset is a sample small one that is used only to investigate the functionality of ID3 Decision Tree classifier.

## Algorithm Implementation

In this project, ID3 decision tree algorithm is coded in python and its functionality is checked on a small dataset sample. In this code, the process starts by encoding the values of an attribute so that it can manipulate both string and numbers in the calculation. Then in each depth of decision tree analysis, the information gain is calculated the attribute with maximum information gain value is chosen for data split.

The process is continued until one of the following stop criteria is reached:

1. If all the attribute_value_pairs are considered in the split process.

2. If the depth of tree reached a predefined value.

3. If all the labels in the split data is pure and so the algorithm returns the value of y as label.

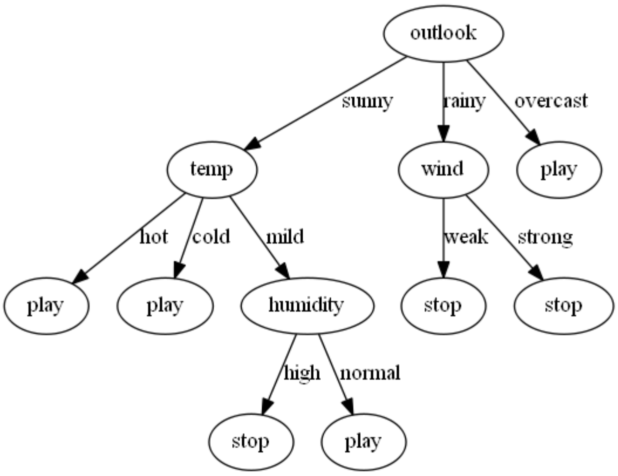
The final Decision tree is stored as nested dictionaries. Below you can see a sample of a binary decision tree in the form of a nested dictionary:

```
{('outlook', 'sunny'):
         {('temp', 'hot'): 'play',
          ('temp', 'mild'):
                    {('humidity', 'high'): 'stop',
                     ('humidity', 'normal'): 'play'},
           ('temp', 'cold'): 'play'},
  ('outlook', 'rainy'):
          {('wind', 'weak'): 'stop',
           ('wind', 'strong'): 'stop'},
 ('outlook', 'overcast'): 'play'}
```

The decision tree is then plotted in the python console similar to the following sample. In this way the user can observe how decision tree is formed.

```
TREE
+---[@@@@: outlook = sunny ]
|         +---[@@@@: temp = hot ]
|         |         +---[LABEL ===> play]
|         +---[@@@@: temp = mild ]
|         |         +---[@@@@: humidity = high ]
|         |         |         +---[LABEL ===> stop]
|         |         +---[@@@@: humidity = normal ]
|         |         |         +---[LABEL ===> play]
|         +---[@@@@: temp = cold ]
|         |         +---[LABEL ===> play]
+---[@@@@: outlook = rainy ]
|         +---[@@@@: wind = weak ]
|         |         +---[LABEL ===> stop]
|         +---[@@@@: wind = strong ]
|         |         +---[LABEL ===> stop]
+---[@@@@: outlook = overcast ]
|         +---[LABEL ===> play]
```

Finally, to better visualize the process of prediction in tree model, a figure which is plotting the nodes and leaves of the tree is drawn using graphviz package. To do so, firstly the dot_string source data is composed for graphviz and then using the dot format, graphviz creates the tree model and saves it as *Created_Tree.png*. The figure below depicts a graph generated by graphviz for a sample ID3 decision tree.


outlook
sunny
rainy
overcast
temp
wind
play
hot
cold
mild
weak
strong
play
play
humidity
stop
stop
high
normal
stop
play