PROJECT REPORT

TOPIC: Implementation Of Decision Tree On Play Tennis Dataset And Comparison On Entropy & Gini Index

**BY: RAJAS V PRASAD**

**BRANCH: MEDICAL ELECTRONICS**

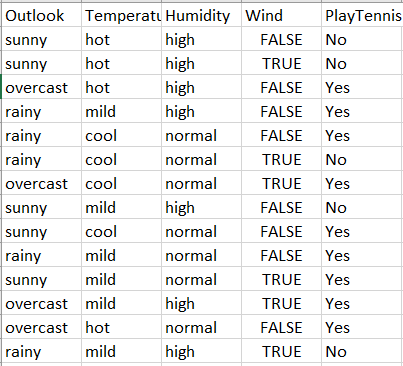
**Abstract:**

The dataset consists of Columns like: Outlook, Temperature, Humidity, Wind and PlayTennis. This dataset gives the possibility of playing tennis when features like Outlook, Temperature, Humidity and Wind are taken into consideration. From the considered features we visualize a decision tree. The platform and language used to build the model is ‘Jupyter Notebook’ and Python 3.0 respectively. The process is elaborated below.

**Steps to implement in our model:**

1. Importing the data
2. Preparing the data
3. Visualizing the Decision tree

The dataset under reference is created in Excel Sheet. The image below depicts the dataset we have considered.

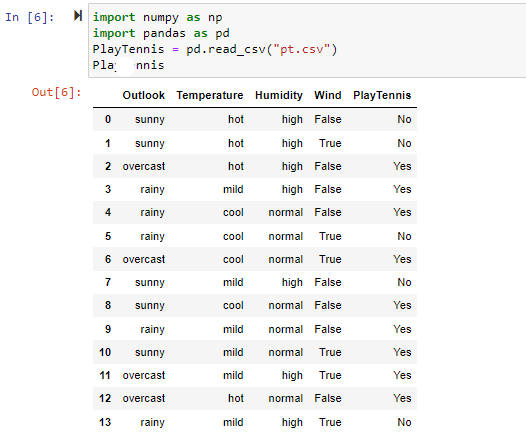


This dataset is saved as a comma separated value file aka ‘csv’ File.

Step 1)

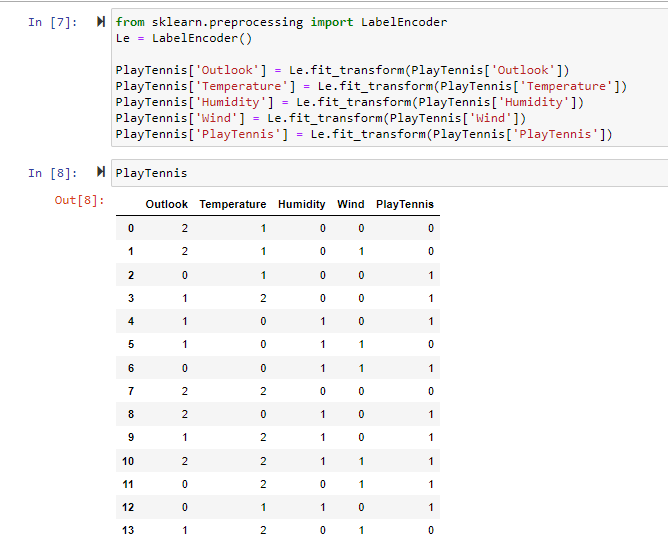
*IMPORTING THE DATA*

The playtennis dataset(pt.csv) was imported into jupyter notebook using ‘pandas’. The data was understood by the machine by the call ‘pd.read\_csv(“pt.csv”)’



The data was then transformed to labels in depiction of ‘numbers’ by usage of library called ‘LabelEncoder()’

The respective columns (Outlook, Temperature, Humidity, Wind, PlayTennis) were selected and each of the columns categorical features were converted to number depiction.



From the above image…

In Outlook column value ‘2’ refers ‘sunny’, value ‘0’ refers ‘overcast’ and value ‘1’ refers ‘rainy’.

In Temperature column value ‘1’ refers ‘hot’, value ‘2’ refers ‘mild’ and value ‘0’ refers ‘cool’.

In Humidity column value ‘0’ refers ‘high’ and value ‘1’ refers ‘normal’.

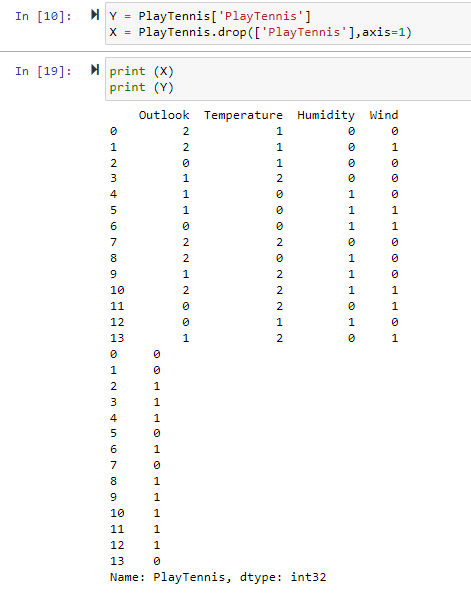
In Wind column value ‘0’ refers ‘False’ and value ‘1’ refers ‘True’.

Step 2):

*PREPARING THE DATA*

To prepare the data we first need to split the files to make sure the machine understands that ‘PlayTennis’ column is the decision column taken under and the rest of the columns are defining the decision.

For that purpose we use ‘.drop’ by choosing the columns.



So now we have X to be having the columns ‘Outlook’, ‘Humidity’, ‘Temperature’ and ‘Wind’.

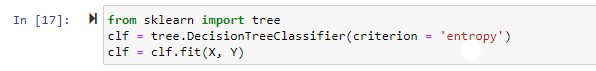
Y to be having the column ‘PlayTennis’.

Step 3):

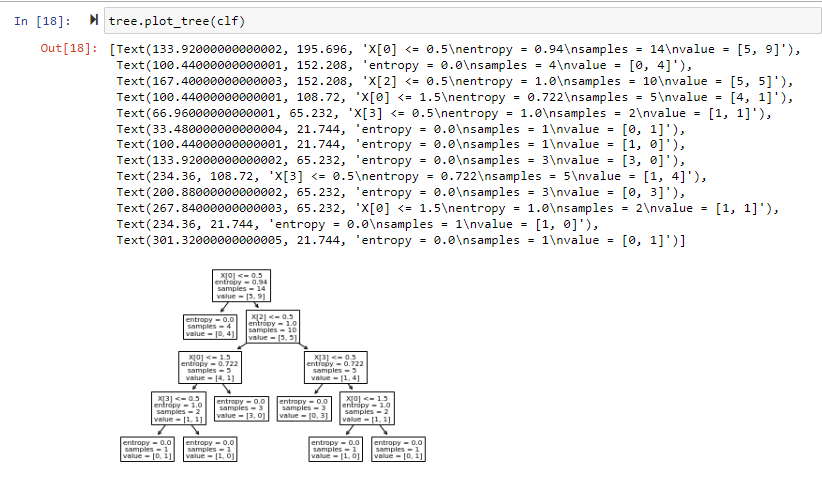
*VISUALIZING THE DECISION TREE*

Using the ‘DecisionTreeClassifier’ library from scikit will help the machine to implement a decision tree based on the given dataset.

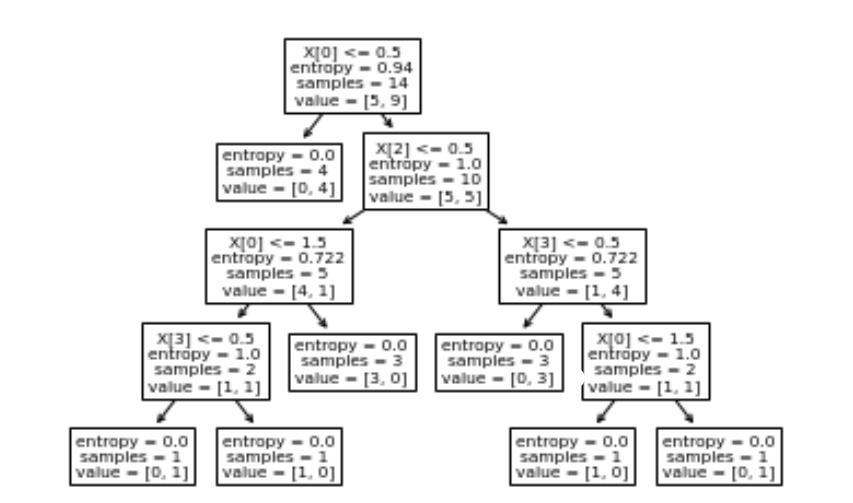
So let us first visualize for the criterion ‘entropy’ and fit the variables ‘X’ & ‘Y’ wrt to ‘DecisionTreeClassifier’.



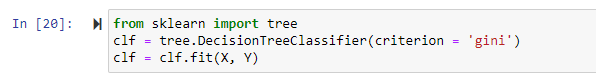
To visualize the tree we use ‘tree.plot\_tree(clf)’

what is happening here?

The model is finding Entropy for each of these columns i.e. Outlook, Temperature, Humidity & Wind. Then it is finding the information gain wrt to each of these columns. The maximum information gain from each of these coulmns will give the first split. This continues by finding the highest info gain as spiltting takes place.



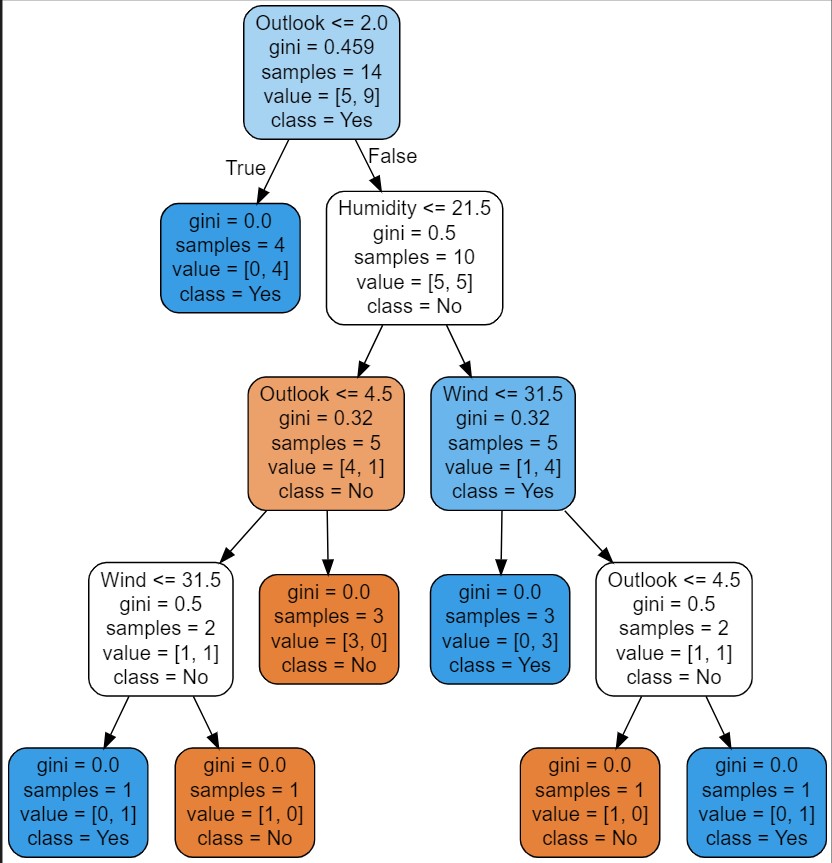
Next criterion will be for using the “DecisionTreeClassifier’ for the criterion ‘gini’ that refers to Gini Index.



Then we export the decision tree as a dot file and visualize it in visual studio code.



Once the dot file is created we open it to visualize our new decision tree.**[Note: In VSCODE open extensions and search ‘Graphviz Preview by EFanZh’. Then open command palette(ctrl+shift+P) and then select ‘Graphviz open preview to the side’ for the existing code being selected]**



The above decision tree works like…

Finding the gini index for each of these columns: Windy, Humidity, Temperature and Outlook. **The formula is : [ 1 – {P1}^2 – {P2}^2 ]**

For windy we promote gini index for False & True and then we find the average gini index of windy.

In the same manner the gini index is calculated for Humidity, Outlook & Temperature. Once this is done The minimum gini index is taken into consideration and performed with the first split.

**THANK YOU…**