Decision trees

Farwa Ahmed p136097

Modeling iris data set using decision trees

Abstract:

Decision Trees model has two goals: producing an accurate classifier and understanding the predictive structure of the problem. They are used in many different disciplines including diagnosis, cognitive science, artificial intelligence, game theory, engineering and data mining.

Introduction:

1. A tree like model is drawn in an upside down manner.
2. We can split the nodes at each level by starting from the root node.
3. Each internal node splits the instance space into two or more subspaces according to a certain discrete function of the input attributes values.
4. We keep on splitting until the leaf node is reached which represents the particular decision or the outcome.
5. A decision tree is a classifier expressed as a recursive partition of the instance space.
6. Usually, the tree complexity is measured by one of the following metrics: the total number of nodes, total number of leaves, tree depth and number of attributes used.

Advantages

1. With the help of decision trees you can handle missing values
2. You can handle errors easily in your dataset with the help of decision trees.
3. Categorical and numerical data both can be handled with these kind of trees.
4. They use natural selection method for feature selection

Disadvantages

1. Not good for unbalanced classes
2. Small variation of data leads to substantially different trees.
3. Overfitting can occur

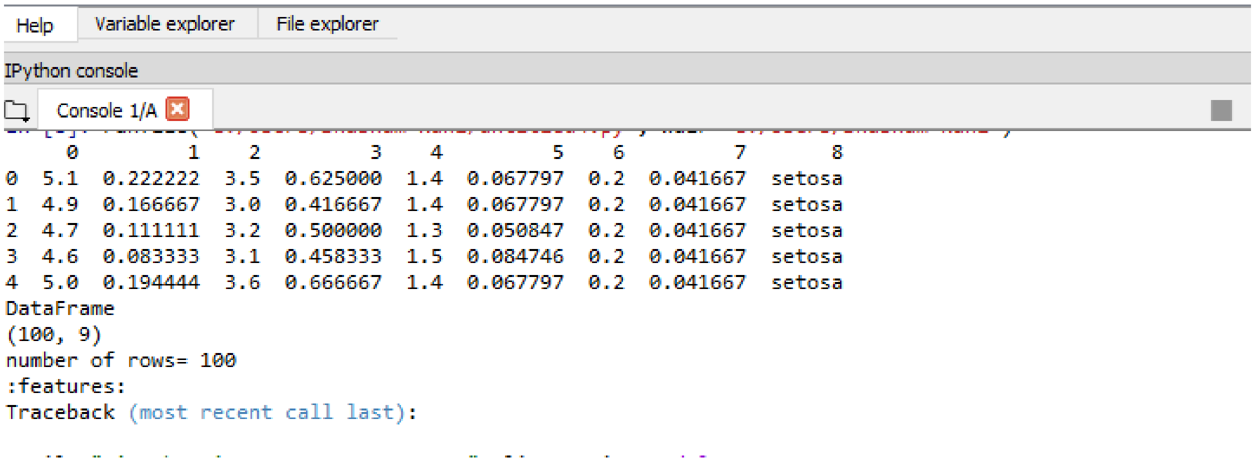
Conclusions:

Decision trees are known as highly efficient tools of machine learning and data mining, capable to produce accurate and easy-to-understand models. They are robust and perform well with large data in short time. They are very efficient predictive model.

Github link:

https://gist.github.com/Farwaa/25e157285a40507c432ba5cd215410d5

Screenshots:



```
     0         1    2         3    4         5    6         7       8
0  5.1  0.222222  3.5  0.625000  1.4  0.067797  0.2  0.041667  setosa
1  4.9  0.166667  3.0  0.416667  1.4  0.067797  0.2  0.041667  setosa
2  4.7  0.111111  3.2  0.500000  1.3  0.050847  0.2  0.041667  setosa
3  4.6  0.083333  3.1  0.458333  1.5  0.084746  0.2  0.041667  setosa
4  5.0  0.194444  3.6  0.666667  1.4  0.067797  0.2  0.041667  setosa
DataFrame
(100, 9)
number of rows= 100
:features:
Traceback (most recent call last):
```