

Decision Tree in Sleep Stage Prediction Model

Architecture

The initial decision tree was a default primitive decision tree offered by scikit-learn library of python. The default parameters for the architecture of the decision tree are shown as follows :-

- **criterion : optional (default="gini") or Choose attribute selection measure:** This parameter allows us to use the different-different attribute selection measure. Supported criteria are "gini" for the Gini index and "entropy" for the information gain.
- **splitter : string, optional (default="best") or Split Strategy:** This parameter allows us to choose the split strategy. Supported strategies are "best" to choose the best split and "random" to choose the best random split.
- **max_depth : int or None, optional (default=None) or Maximum Depth of a Tree:** The maximum depth of the tree. If None, then nodes are expanded until all the leaves contain less than min_samples_split samples. The higher value of maximum depth causes overfitting, and a lower value causes underfitting ([Source](#)).

The parameters used in our model are :-

```
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion='entropy',max_depth=12)
```

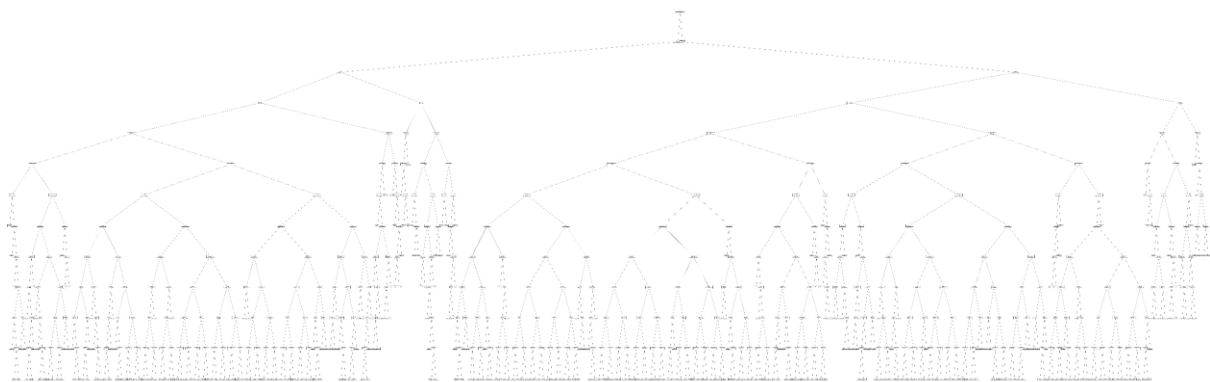
The {criterion='entropy'} splits the data-set based on the weighted entropy of each attribute, in our case, the weighted entropy of each signal after normalizing.

The {max_depth=12} restricts the decision tree to a maximum depth of 12. As stated above, a too high value of maximum depth can cause overfitting, and a lower value causes underfitting. A maximum depth of 12 gives an optimum solution, which has been verified through K-fold Cross Validation.

K-fold Cross Validation

We have performed K-fold cross validation to verify whether the model is undergoing overfitting or underfitting. We have first shuffled the dataset randomly. Then we have splitted the dataset into 5 sets/folds. Then we send 4 folds into training and 1 fold into testing. This is iterated until each fold goes into testing. K-fold cross validation has been very aptly and concisely demonstrated in the diagram given below.





The above diagram gives an idea of how the decision tree looks. The picture is available in the github repository, for reference.

Classification Report

	precision	recall	f1-score	support
0	0.95	0.98	0.96	7621
1	0.84	0.69	0.76	444
2	0.76	0.73	0.75	1834
3	0.75	0.81	0.78	1632
4	0.82	0.64	0.72	969
accuracy			0.88	12500
macro avg	0.82	0.77	0.79	12500
weighted avg	0.88	0.88	0.88	12500

Github Repository

<https://github.com/Jibitesh-Chakraborty2811/Sleep-Stage-Decision-Tree>