Artificial Intelligence: Lab 4

Supervised Machine Learning : Decision Trees & Random Forests

1 Decision Trees

A decision tree is a tree structure where an internal node represents a feature, the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It learns to partition on the basis of the attribute value. It partitions the tree in recursively manner call recursive partitioning. This tree structure helps in decision making. It's visualisation like a flowchart diagram which easily mimics the human level thinking. That is why decision trees are easy to understand and interpret.

For this lab, we will use pima-indians-diabetes dataset. It is well described in the following address :

`https://www.kaggle.com/kumargh/pimaindiansdiabetescsv`

1. Import the necessary python modules : `numpy`, `pandas` and `matplotlib`
2. Load the dataset. It is available at the address :

   `https://www.labri.fr/~zemmari/datasets/pima-indians-diabetes.csv`

3. In the sequel, we will not consider entire the dataset. Write instructions to extract the columns `pregnant`, `insulin`, `bmi`, `age`, `glucose`, `bp`, and `pedigree`.
4. Split the dataset into two subsets : one for training and the other for testing. Save 30% of the dataset for test.
5. Using the python module `sklearn.tree`, train a model using `DecisionTreeClassifier`.
6. Give the confusion matrix and evaluate the model.
7. You can visualise the trained tree. For this purpose, you can execute the following instructions :

```
from sklearn.tree import export_graphviz
from sklearn.externals.six import StringIO
from IPython.display import Image
import pydotplus

dot_data = StringIO()
export_graphviz(dt, out_file=dot_data,
            filled=True, rounded=True,
            special_characters=True,feature_names = feature_cols,class_names=['0','1'])
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('diabetes.png')
Image(graph.create_png())
```

2 Random Forests

Random forests is a supervised learning algorithm. It is the most flexible and easy to use algorithm. A forest is comprised of trees. It is said that the more trees it has, the more robust a forest is. Random forests creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a pretty good indicator of the feature importance.

1. Using `RandomForestClassifier` from the python module `sklearn.ensemble`, train a model using the same dataset and evaluate its accuracy. What do you observe ?
2. Observe the set of parameters and their default values used to train your model. Try other values for these parameters and observe the obtained accuracies.
3. Use a random grid and random search to find the best parameters for your classifier.
4. Train and evaluate your model.