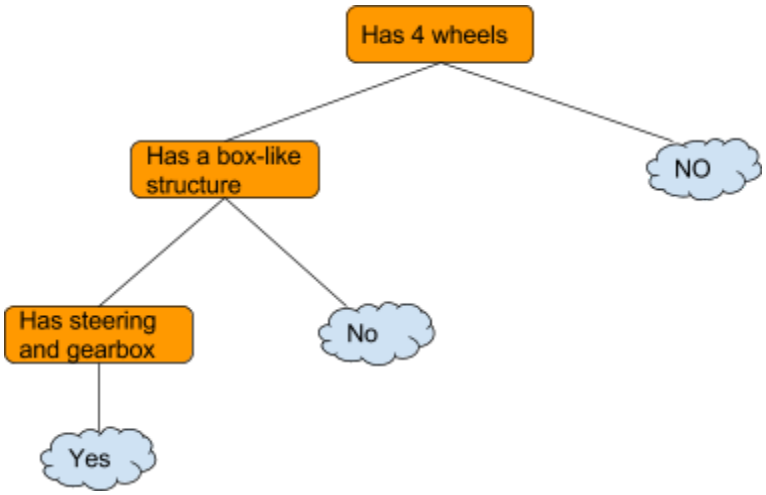
# Decision Tree classifier - Explanation & Example using Iris dataset

Hey, you may want to check out on https://codingmachinelearning.wordpress.com/2016/06/16/blog-post-5-singular-value-decomposition/ to recap about using svd for dimensionality reduction.

In this post we will deal with Decision Tree Classifier(DTC). DTC can be simply assumed as a series of if-else questions asked at each decision node. For example if I need to know a given object is a car or not, I will ask a series of questions which can be interpreted as if-else statements. I ask, if the object had 4 tyres. If yes then does it have a box-like structure around it. If yes then does it have a steering-wheel and gear box. If answer to all of these decision questions is yes probably the object is car. Else it may not be a car.

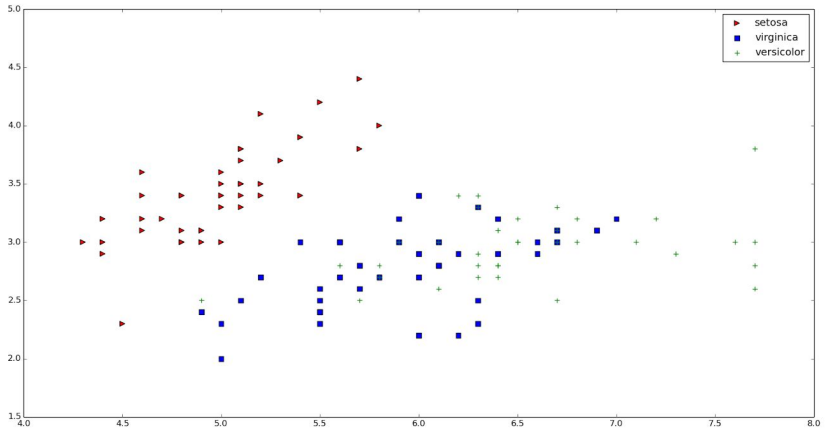



This is a very small example of a decision tree. This is used only for illustration purpose and I hope it drives the message that DTC is nothing but a series of if-else situations at each decision node.

Now we will look into the iris data set and try to implement decision tree classifier algorithm on the same. By implement, most the classifiers have been coded and integrated in scikit sklearn package. We will all we need by using sklearn

Iris data set contains details about different flowers. Based on the features we need to be able to predict the flower type. Please find the description of iris data set here

- First step is to load the iris data set into variables x and y where x contains the data (4 columns) and y contains the target.
- Split the data into train and validation set so that we can see the performance of our model on “previously unseen” instances
- Sepal length,Sepal width, Petal length, Petal width are its features and the targets are setosa (0), Iris virginica(1) and Iris versicolor(2)
- Visualize the data set to see how the points are spread in space




This is the plot we obtain by plotting the first 2 feature points (of sepal length and width)

Now we invoke sklearn decision tree classifier to learn from iris data. This simply translates to the following code.

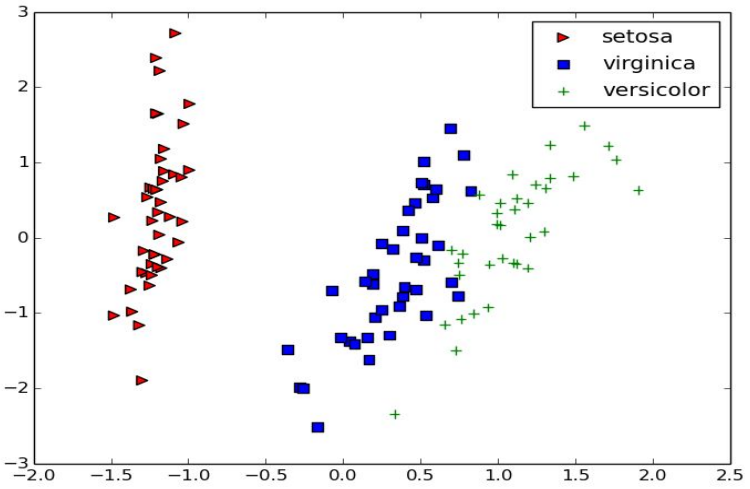
```
clf=DecisionTreeClassifier()

clf.fit(train,train_lab)

output=clf.predict(test)
```

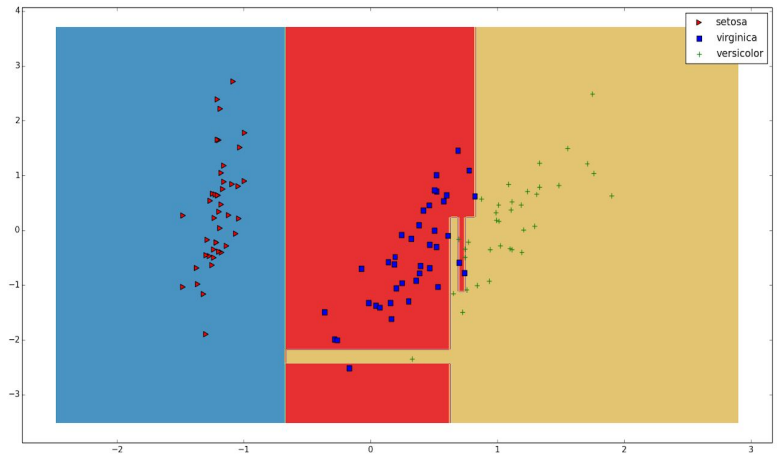
This piece of code, creates an instance of Decision tree classifier and fit method does the fitting of the decision tree.

We then invoke predict method to be able to get the prediction for the test-set where our accuracy is close to 90% (on the 30 test instances)

We can also aim to plot the decision surface for the iris data set and see how decision tree classifiers learn rectangular boundaries. The pair wise plots for the iris data set is clearly provided in the sklearn. Link provided at the end of the article.

Our aim here is to perform PCA on the data set, project data on its principle components and reduce data to 2 dimensions. The visualization of the above task is given below and the code for the same will be made freely available on github.




This is the PCA plot of the data and that now we can focus on obtaining the decision surface of the classifier we have built. Also we can get the tree-type structure for the same using the dot image from sklearn.

We now plot the decision surface for the same.

We can see clearly the rectangular decision boundary learned by our classifier. If a point falls in the blue surface it will be classified as setosa(0) and so on.

**Things to remember:**

- Decision tree gives rectangular decision boundaries
- It can be thought of as series of if-else questions at each decision node

*In the next post we will look into learning SVM and the kind of decision boundary it can generate. Till then bye!!*

Code : https://github.com/vsuriya93/coding-machine-learning/tree/master/decision-tree
Sklearn iris visualization: http://scikit-learn.org/stable/auto_examples/tree/plot_iris.html