**Experiment No 7**

**Title:** To Implement random forest classifier

**Tools Required: Anaconda Navigator**

**Concept :**

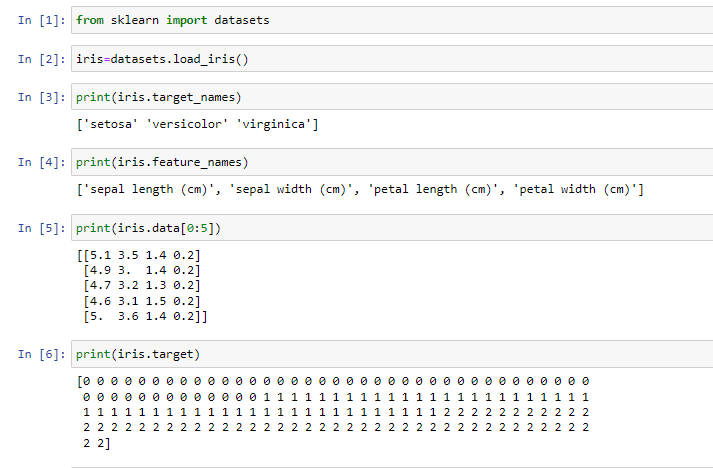
Random forests is a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest is comprised of trees. It is said that the more trees it has, the more robust a forest is. Random forests creates decision trees on 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

**Example Problem**

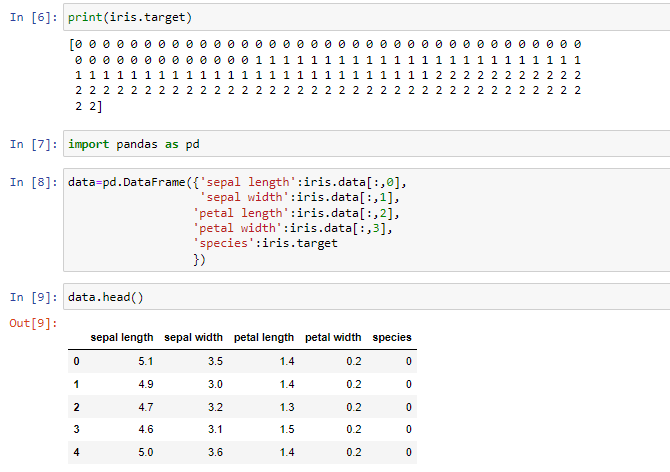
You will be building a model on the iris flower dataset, which is a very famous classification set. It comprises the sepal length, sepal width, petal length, petal width, and type of flowers. There are three species or classes: setosa, versicolor, and virginia. You will build a model to classify the type of flower. The dataset is available in the scikit-learn library or you can download it from the UCI Machine Learning Repository.

**Steps:**

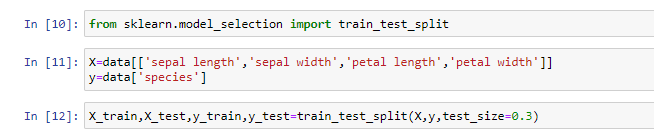
1. Import the required libraries for our random forest analysis & pull in the required data



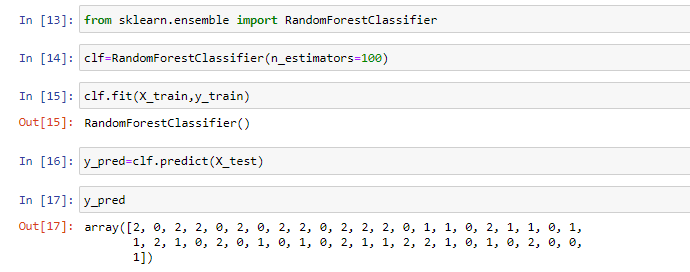
1. After loading the data, we understand the structure & variables, determine the target & feature variables (dependent & independent variables respectively)



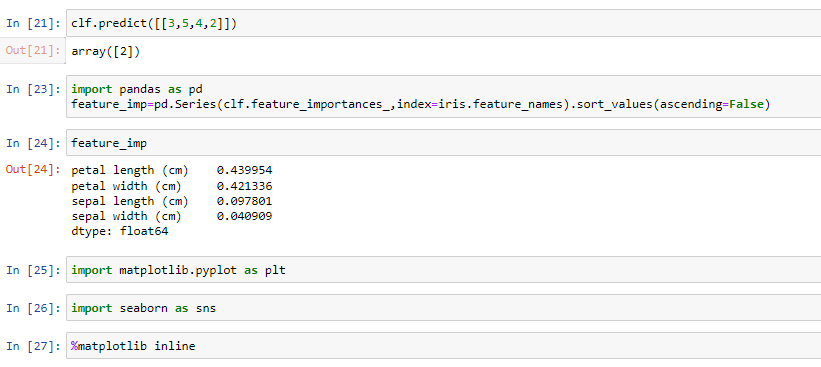
1. Splitting of data into training & testing sets in the ratio of 70:30.

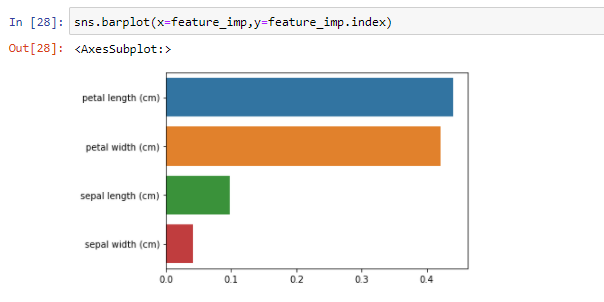


1. Performing random forest analysis using scikit learn
   1. Create random forest classifier object
   2. Train random forest Classifier
   3. Predict the response for test dataset

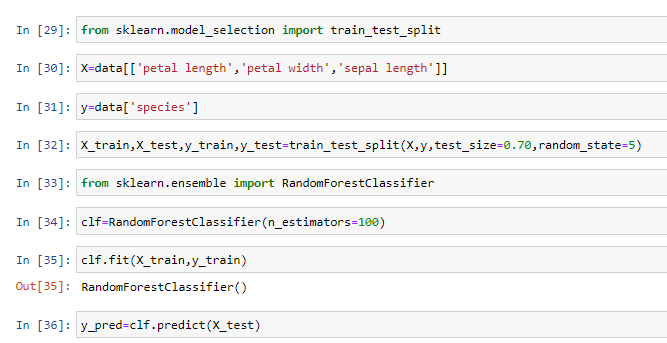


1. We should estimate how accurately the classifier predicts the outcome. The accuracy is computed by comparing actual test set values and predicted values.





1. Finding Important Features in Scikit-learn
   1. First, you need to create a random forests model.
   2. Second, use the feature importance variable to see feature importance scores.
   3. Third, visualize these scores using the seaborn library.



1. Find the accuracy of newly formed forest classifier with important features.

