# 10. Random Forest Classifier

November 15, 2022

## 1 Random Forest Classifier

[1]:
```
# importing required libraries
# importing Scikit-learn library and datasets package
from sklearn import datasets

# Loading the iris plants dataset (classification)
iris = datasets.load_iris()
```

[2]:
```
print(iris.target_names)
```

```
['setosa' 'versicolor' 'virginica']
```

[3]:
```
print(iris.feature_names)
```

```
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width
(cm)']
```

[4]:
```
# dividing the datasets into two parts i.e. training datasets and test datasets
X, y = datasets.load_iris(return_X_y = True)

# Spliting arrays or matrices into random train and test subsets
from sklearn.model_selection import train_test_split

# i.e. 70 % training dataset and 30 % test datasets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30)
```

[5]:
```
# importing random forest classifier from assemble module
from sklearn.ensemble import RandomForestClassifier
import pandas as pd

# creating dataframe of IRIS dataset
data = pd.DataFrame({'sepallength': iris.data[:, 0], 'sepalwidth': iris.data[:,␣
 ↪1],'petallength': iris.data[:, 2],
                     'petalwidth': iris.data[:, 3],'species': iris.target})
```

[6]:
```
# printing the top 5 datasets in iris dataset
print(data.head())
```

```
   sepallength  sepalwidth  petallength  petalwidth  species
0          5.1         3.5          1.4         0.2        0
1          4.9         3.0          1.4         0.2        0
2          4.7         3.2          1.3         0.2        0
3          4.6         3.1          1.5         0.2        0
4          5.0         3.6          1.4         0.2        0
```

[7]:

```
# creating a RF classifier
clf = RandomForestClassifier(n_estimators = 100)

# Training the model on the training dataset
# fit function is used to train the model using the training sets as parameters
clf.fit(X_train, y_train)

# performing predictions on the test dataset
y_pred = clf.predict(X_test)

# metrics are used to find accuracy or error
from sklearn import metrics
print()

# using metrics module for accuracy calculation
print("ACCURACY OF THE MODEL: ", metrics.accuracy_score(y_test, y_pred))
```

```

ACCURACY OF THE MODEL:  0.9333333333333333
```

[8]:

```
# predicting which type of flower it is.
clf.predict([[3, 3, 2, 2]])
```

[8]:

```
array([0])
```

[9]:

```
# importing random forest classifier from assemble module
from sklearn.ensemble import RandomForestClassifier
# Create a Random forest Classifier
clf = RandomForestClassifier(n_estimators = 100)

# Train the model using the training sets
clf.fit(X_train, y_train)
```

[9]:

```
RandomForestClassifier()
```

[10]:

```
# using the feature importance variable
import pandas as pd
feature_imp = pd.Series(clf.feature_importances_, index = iris.feature_names).
 ↪sort_values(ascending = False)
feature_imp
```

```
[10]: petal length (cm)    0.478822
      petal width (cm)     0.431389
      sepal length (cm)    0.070939
      sepal width (cm)     0.018849
      dtype: float64
```