**TITLE: RANDOM FOREST REGRESSION**

**CODE: -**

# -\*- coding: utf-8 -\*-

"""

Created on Wed Feb 27 14:39:20 2019

@author: RoKu

"""

# Random Forest Regression

# Importing the libraries

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

# Importing the dataset

dataset = pd.read\_csv('Dataset\_flwr.csv')

X = dataset.iloc[:, 1:2].values

y = dataset.iloc[:, 2:].values

# Splitting the dataset into the Training set and Test set

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2, random\_state = 0)

# Feature Scaling

from sklearn.preprocessing import StandardScaler

sc\_X = StandardScaler()

X\_train = sc\_X.fit\_transform(X\_train)

X\_test = sc\_X.transform(X\_test)

sc\_y = StandardScaler()

y\_train = sc\_y.fit\_transform(y\_train)

# Fitting the Random Forest Regression to the dataset

from sklearn.ensemble import RandomForestRegressor

regressor = RandomForestRegressor(n\_estimators = 300, random\_state = 0)

regressor.fit(X, y)

# Predicting a new result

y\_pred = regressor.predict(y\_test)

# Visualising the Regression results (for higher resolution and smoother curve)

X\_grid = np.arange(min(X), max(X), 0.01)

X\_grid = X\_grid.reshape((len(X\_grid), 1))

plt.scatter(X, y, color = 'red')

plt.plot(X\_grid, regressor.predict(X\_grid), color = 'green')

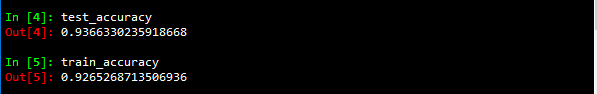
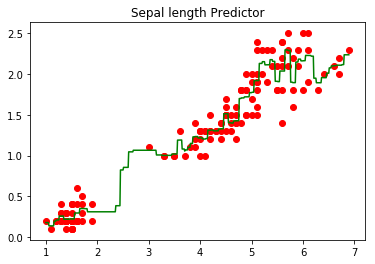
plt.title('Sepal length Predictor')

plt.xlabel = ('Petal Length')

plt.ylabel = ('Species')

plt.show()

**RESULT:**



**ACCURACIES**

**TITLE: Decision Tree REGRESSION**

**CODE: -**

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

# Importing the dataset

dataset = pd.read\_csv('Dataset\_flwr.csv')

X = dataset.iloc[:, 0:1].values

y = dataset.iloc[:, 1:].values

# Splitting the dataset into the Training set and Test set

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2, random\_state = 0)

# Feature Scaling

from sklearn.preprocessing import StandardScaler

sc\_X = StandardScaler()

X\_train = sc\_X.fit\_transform(X\_train)

X\_test = sc\_X.transform(X\_test)

sc\_y = StandardScaler()

y\_train = sc\_y.fit\_transform(y\_train)

y\_test = sc\_y.fit\_transform(y\_test)

# Fitting the Decision Tree Regression to the dataset

from sklearn.tree import DecisionTreeRegressor

regressor = DecisionTreeRegressor(random\_state = 0)

regressor.fit(X, y)

# Predicting a new result

y\_pred = regressor.predict(y\_test)

# Visualising the Regression results

plt.scatter(X, y, color = 'red')

plt.plot(X, regressor.predict(X), color = 'blue')

plt.title('Sepal length Predictor')

plt.xlabel = ('Petal Length')

plt.ylabel = ('Species')

plt.show()

# Visualising the Regression results (for higher resolution and smoother curve)

X\_grid = np.arange(min(X), max(X), 0.1)

X\_grid = X\_grid.reshape((len(X\_grid), 1))

plt.scatter(X, y, color = 'red')

plt.plot(X\_grid, regressor.predict(X\_grid), color = 'blue')

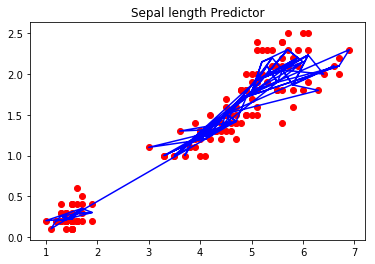
plt.title('Sepal length Predictor')

plt.xlabel = ('Petal Length')

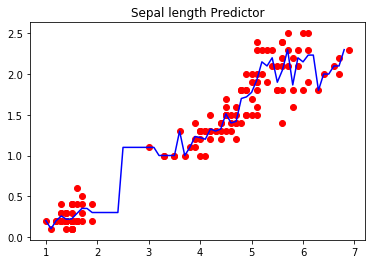
plt.ylabel = ('Species')

plt.show()

**RESULT:**



**(PLOT without NP arrange)**



**(plot with arrange)**

|  |  |
| --- | --- |
| **SUBMISSION DATE**-  **15-02-19** | **SIGN OF COURSE INSTRUCTOR**- |
| **ROLL NO OF THE STUDENT-**  **01** | **NAME OF THE STUDENT-**  **Rohit Kulkarni** |