RANDOM FOREST REGRESSION

Mohan M J

INTRODUCTION

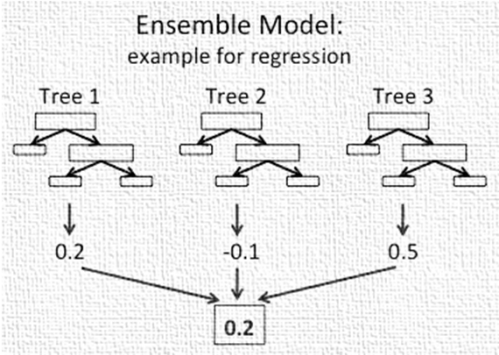
STEP 1: Pick random K data points from the training set

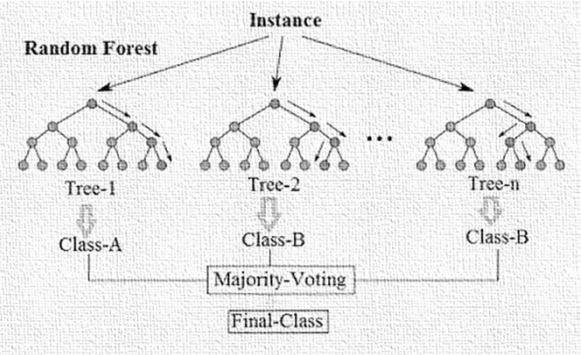
STEP 2: Build the Decision Tree associated to these K data points

STEP 3: Choose the number Ntree of trees you want to build and repeat STEPS 1 & 2

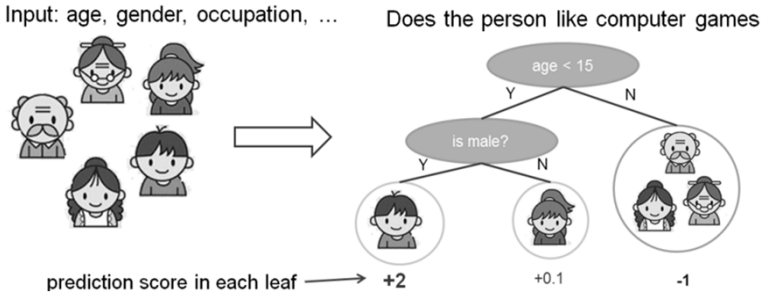
STEP 4: For a new data point make each one of Ntree trees predict the value of Y for the data point in question. The new data point is the average across all of predicted Y values

INTRO

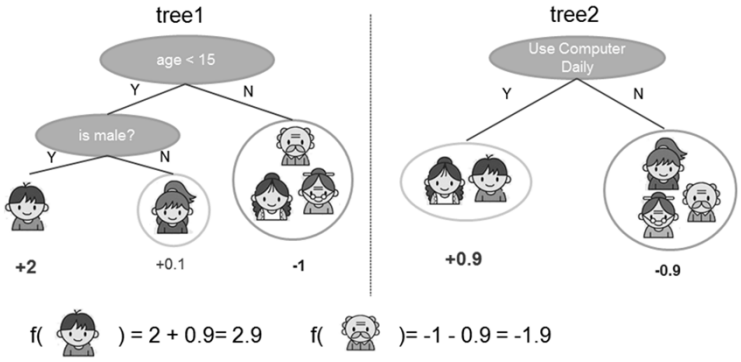

Ensemble Model:
example for regression
Tree 1
Tree 2
Tree 3
0.2
-0.1
0.5
0.2


Instance
Random Forest
Tree-1
Tree-2
Tree-n
Class-A
Class-B
Class-B
Majority-Voting
Final-Class

INTRO


Input: age, gender, occupation, …
Does the person like computer games
age < 15
Y
N
is male?
Y
N
prediction score in each leaf
+2
+0.1
-1

INTRO



f( ) = 2 + 0.9= 2.9 f( )= -1 - 0.9 = -1.9

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F}$$

where *K* is the number of trees, *f* is a function in the functional space *F*, and *F* is the set of all possible CARTs.

PYTHON CODE:

```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
# Importing the dataset
dataset = pd.read_csv('Position_Salaries.csv')
dataset
X = dataset.iloc[:, 1:2].values
y = dataset.iloc[:, 2].values
```

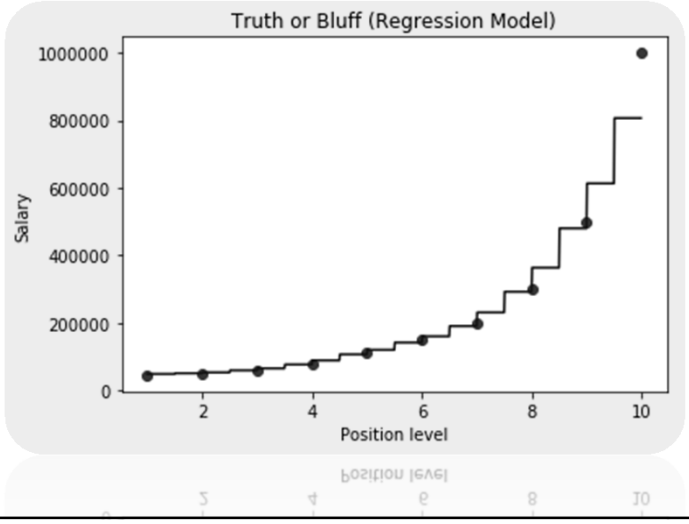
PYTHON CODE:

```
# Fitting the Regression Model 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(6.5)
y_pred
```

PYTHON CODE:

```
# 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 = 'blue')
plt.title('Truth or Bluff (Regression Model)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```

PYTHON CODE:



THANK YOU