



# RANDOM FOREST REGRESSION

Akhilesh  
Joshi

# Random Forest Intuition

---

STEP 1: Pick at 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  $N_{tree}$  of trees you want to build and repeat STEPS 1 & 2



STEP 4: For a new data point, make each one of your  $N_{tree}$  trees predict the value of  $Y$  to for the data point in question, and assign the new data point the average across all of the predicted  $Y$  values.

PYTHON



# READING FILE DYNAMICALLY

```
from tkinter import *  
from tkinter.filedialog import askopenfilename  
  
root = Tk()  
root.withdraw()  
root.update()  
file_path = askopenfilename()  
root.destroy()
```

# IMPORTING LIBRARIES

```
import pandas as pd
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

# IMPORTING DATASET

```
dataset = pd.read_csv(file_path)
```

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

```
y= dataset.iloc[:,2:3].values
```

# RANDOM FOREST TREEZ REGRESSOR

```
from sklearn.ensemble import RandomForestRegressor  
regressor = RandomForestRegressor(n_estimators=100,  
random_state = 0)  
  
model = regressor.fit(X,y)
```

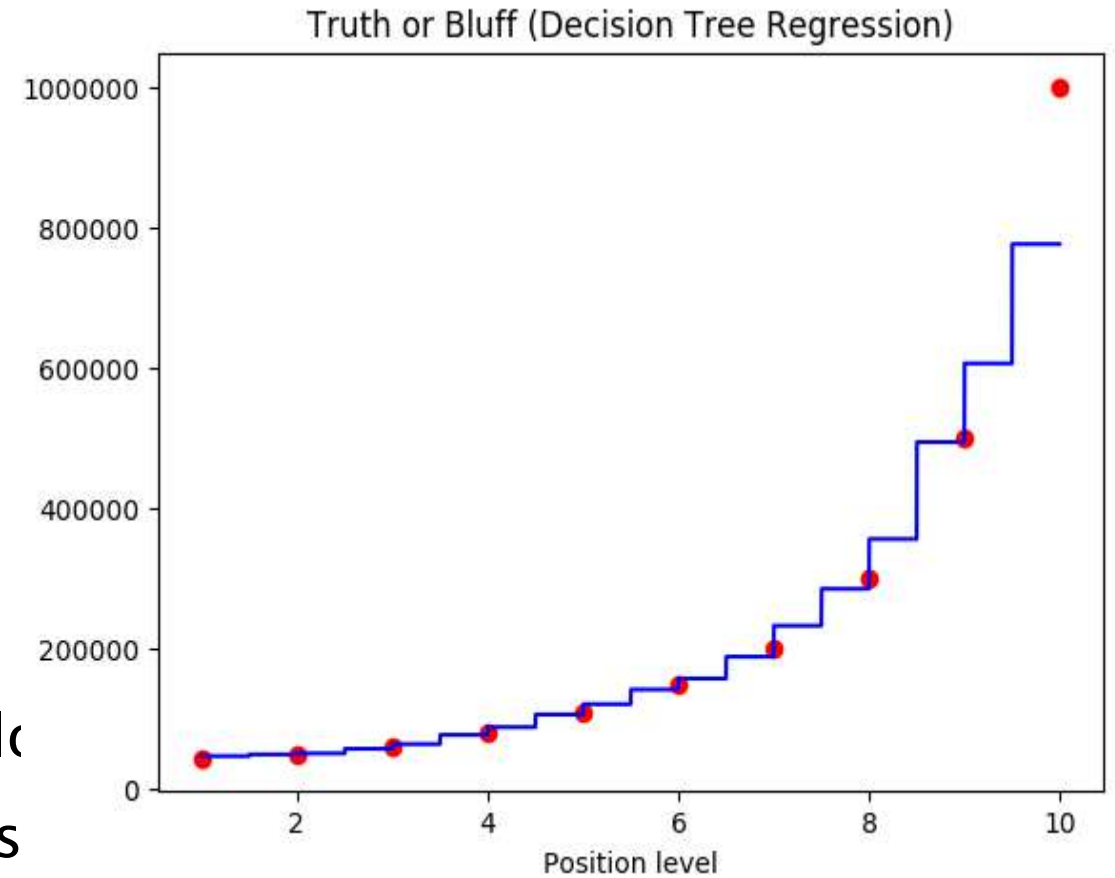
# PREDICTION

```
model.predict(6.5)
```



# PLOT

```
X_grid = np.arange(min(X), max(X), 0.001)
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 (Decision Tree Regression)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```



Note : as number of trees increases you don't see much of increase in steps as more trees start converging

R



# READ DATASET

```
library(readr)
```

```
dataset <- read_csv("D:/machine learning AZ/Machine Learning A-Z  
Template Folder/Part 2 – Regression/Section 7 – Support Vector  
Regression (SVR)/SVR/Position_Salaries.csv")
```

```
dataset= dataset[2:3]
```

# LIBRARY REQUIRED - RANDOMFOREST

```
library('randomForest')
```

```
set.seed(1234)
```

```
regressor = ra
```

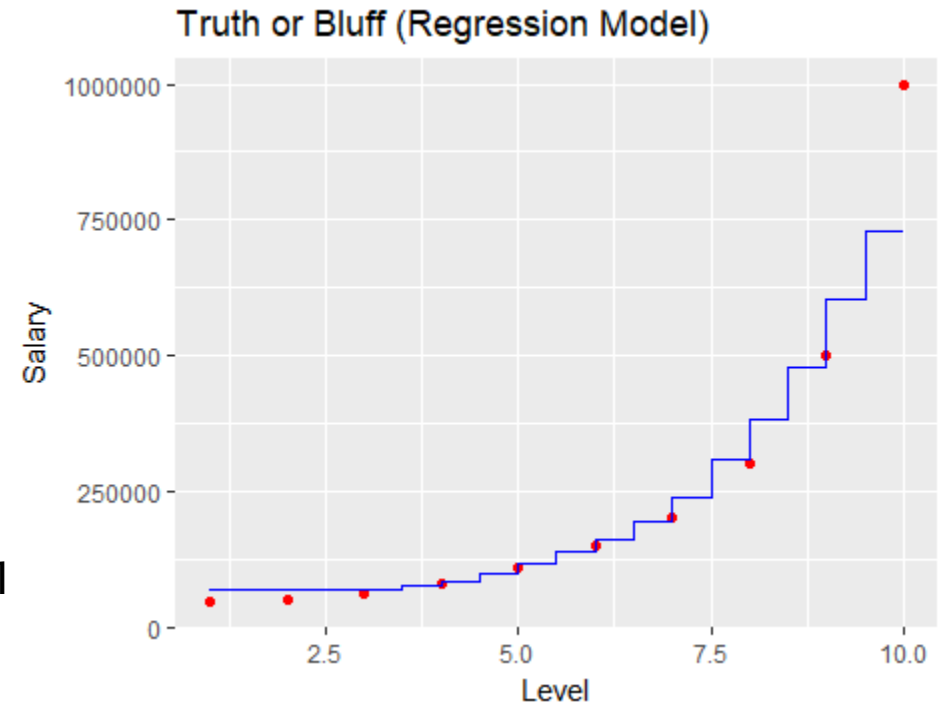
NOTE:

```
#dataset[1] : gives dataframe
```

```
#dataset$Salary : gives a vector since we require vectore for y ndomForest(x=  
dataset[1] , y= dataset$Salary, ntree=1000)
```

# PLOT

```
# install.packages('ggplot2')
library(ggplot2)
x_grid = seq(min(dataset$Level), max(dataset$Level), 0.001)
ggplot() +
  geom_point(aes(x = dataset$Level, y = dataset$Salary),
             colour = 'red') +
  geom_line(aes(x = x_grid, y = predict(regressor, newdata = data.frame(Level = x_grid))),
            colour = 'blue') +
  ggtitle('Truth or Bluff (Regression Model)') +
  xlab('Level') +
  ylab('Salary')
```



# PREDICTION

```
prediction = predict(regressor,data.frame(Level=6.5))
```