

Random Forests Regressions

Ensemble Learning

- Is when we take multiple algorithm or same algorithm multiple times to create something powerful.
1. Pick at random K data points from Training Set
 2. Build a Decision Tree associated to these K data points
 3. Choose the number of N Trees you want to build and repeat Steps 1 and 2.
 4. For a new data point, make each one of your N Trees, predict the values of Y to for the data point in question and assign the new data point the average across all the predicted Y values.

This improves accuracy as you take the average of many predictions, essentially taking the value from a forest of Decision Trees.

Applying Random Forrest Regression in Python

- **Import Libraries**

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

- **Import Dataset**

```
dataset = pd.read_csv('Position_Salaries.csv')
X = dataset.iloc[:, 1:-1].values
y = dataset.iloc[:, -1].values
```

- **Training the Random Forest Regression model on the whole dataset**

```
from sklearn.ensemble import RandomForestRegressor
regressor = RandomForestRegressor(n_estimators = 10, random_state = 0)
regressor.fit(X, y)
```

We need the number of tree's we want to create, which is created using the parameter, `n_estimators`.

- **Predicting a new result**

```
regressor.predict([[6.5]]) #167000
```

- **Visualizing the Decision Tree Regression results (higher resolution)**

```
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 (Random Forest Regression)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```

