



Random Forest Regression in Python

Read

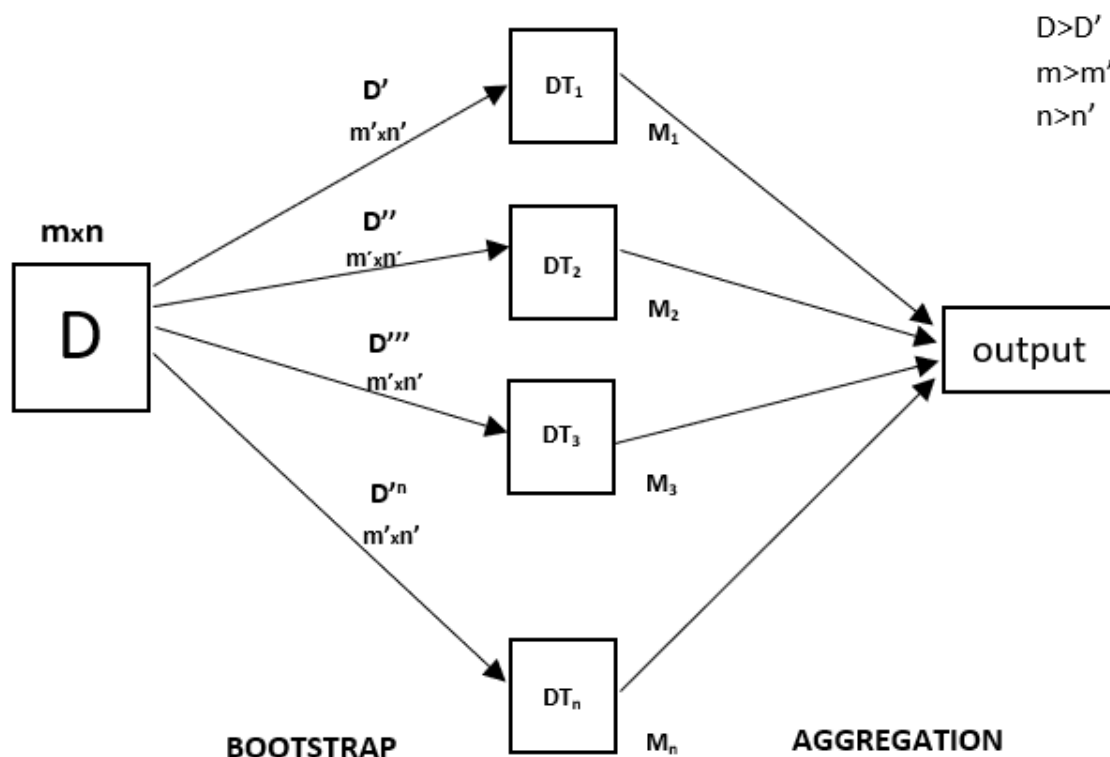
Discuss

Courses

Practice

Video

Every decision tree has high variance, but when we combine all of them together in parallel then the resultant variance is low as each decision tree gets perfectly trained on that particular sample data, and hence the output doesn't depend on one decision tree but on multiple decision trees. In the case of a classification problem, the final output is taken by using the majority voting classifier. In the case of a regression problem, the final output is the mean of all the outputs. This part is called **Aggregation**.



Random Forest Regression Model Working

What is Random Forest Regression?

Random Forest is an ensemble technique capable of performing both



technique called Bootstrap and Aggregation, commonly known as bagging.

The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model. This part is called Bootstrap.

We need to approach the Random Forest regression technique like any other machine learning technique.



- Design a specific question or data and get the source to determine the required data.
- Make sure the data is in an accessible format else convert it to the required format.
- Specify all noticeable anomalies and missing data points that may be required to achieve the required data.
- Create a machine-learning model.
- Set the baseline model that you want to achieve
- Train the data machine learning model.
- Provide an insight into the model with test data
- Now compare the performance metrics of both the test data and the predicted data from the model.
- If it doesn't satisfy your expectations, you can try improving your model

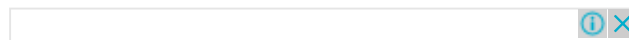
You will be using a similar sample technique in the below example. Below is a step-by-step sample implementation of Random Forest Regression, on the dataset that can be downloaded here- <https://bit.ly/417n3N5>

Import Libraries and Datasets

[Python](#) libraries make it very easy for us to handle the data and perform typical and complex tasks with a single line of code.

- **[Pandas](#)** – This library helps to load the data frame in a 2D array format and has multiple functions to perform analysis tasks in one go.
- **[Numpy](#)** – Numpy arrays are very fast and can perform large computations in a very short time.
- **[Matplotlib/Seaborn](#)** – This library is used to draw visualizations.
- **Sklearn** – This module contains multiple libraries having pre-implemented functions to perform tasks from data preprocessing to model development and evaluation.
- **RandomForestRegressor** – This is the regression model that is based upon the Random Forest model or the ensemble learning that we will be using in this article using the sklearn library.

Python3



```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Now let's load the dataset in the panda's data frame. For better data handling and leveraging the handy functions to perform complex tasks in one go.

Python3

```
data = pd.read_csv('Salaries.csv')
print(data)
```

Output:

	Position	Level	Salary
0	Business Analyst	1	45000
1	Junior Consultant	2	50000
2	Senior Consultant	3	60000
3	Manager	4	80000
4	Country Manager	5	110000
5	Region Manager	6	150000
6	Partner	7	200000
7	Senior Partner	8	300000
8	C-level	9	500000
9	CEO	10	1000000

Select all rows and column 1 from the dataset to x and all rows and column 2 as y.

Python3

```
x = df.iloc[:, :-1]
y = df.iloc[:, -1:]
```

The function enables us to select a particular cell of the dataset, that is, it helps us select a value that belongs to a particular row or column from a set of

Random Forest Regressor Model

Python3

```
# Fitting Random Forest Regression to the dataset
# import the regressor
from sklearn.ensemble import RandomForestRegressor

# create regressor object
regressor = RandomForestRegressor(n_estimators=100,
                                  random_state=0)

# fit the regressor with x and y data
regressor.fit(x, y)
```

Output:

```
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,
                       max_features='auto', max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=1,
                       oob_score=False, random_state=0, verbose=0, warm_start=False)
```

Predicting a new result.

Python3

```
# test the output by changing values
Y_pred = regressor.predict(np.array([6.5]).reshape(1, 1))
```

Now let's visualize the results obtained by using the RandomForest Regression model on our salaries dataset.

Python3

```
# Visualising the Random Forest Regression results

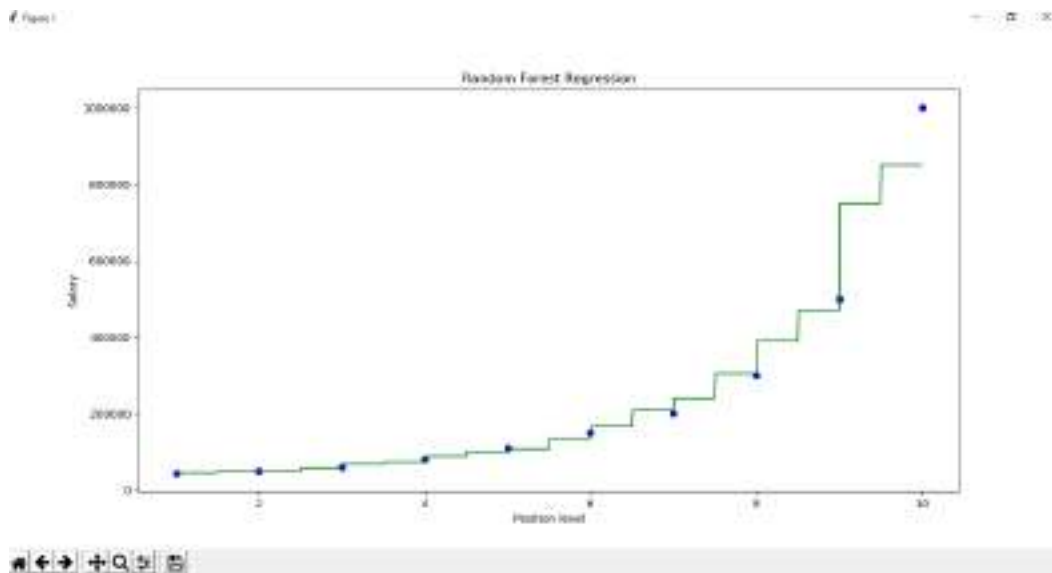
# arrange for creating a range of values
# from min value of x to max
# value of x with a difference of 0.01
# between two consecutive values
X_grid = np.arange(min(x), max(x), 0.01)

# reshape for reshaping the data
# into a len(X_grid)*1 array,
# i.e. to make a column out of the X_grid value
X_grid = X_grid.reshape((len(X_grid), 1))

# Scatter plot for original data
plt.scatter(x, y, color='blue')

# plot predicted data
plt.plot(X_grid, regressor.predict(X_grid),
         color='green')
plt.title('Random Forest Regression')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```

Output:



Out of Bag Score in RandomForest

Bag score or [OOB score](#) is the type of validation technique that is mainly used in bagging algorithms to validate the bagging algorithm. Here a small part of the validation data is taken from the mainstream of the data and the predictions on the particular validation data are done and compared with the other results.

The main advantage that the OOB score offers is that here the validation data is not seen by the bagging algorithm and that is why the results on the OOB score are the true results that indicated the actual performance of the bagging algorithm.

To get the OOB score of the particular Random Forest algorithm, one needs to set the value “True” for the OOB_Score parameter in the algorithm.

Python3

```
from sklearn.ensemble import RandomForestClassifier
RandomForest = RandomForestClassifier(oob_score=True)
RandomForest.fit(X_train,y_train)
print(RandomForest.oob_score_)
```

- It is easy to use and less sensitive to the training data compared to the decision tree.
- It is more accurate than the [decision tree](#) algorithm.
- It is effective in handling large datasets that have many attributes.
- It can handle missing data, [outliers](#), and noisy features.

Disadvantages Random Forest Regression



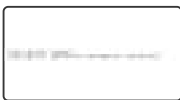
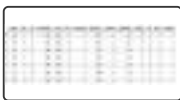






- The model can also be difficult to interpret.
- This algorithm may require some domain expertise to choose the appropriate parameters like the number of decision trees, the maximum depth of each tree, and the number of features to consider at each split.
- It is computationally expensive, especially for large datasets.
- It may suffer from [overfitting](#) if the model is too complex or the number of decision trees is too high.

Whether you're preparing for your first job interview or aiming to upskill in this ever-evolving tech landscape, [GeeksforGeeks Courses](#) are your key to success. We provide top-quality content at affordable prices, all geared towards accelerating your growth in a time-bound manner. Join the millions we've already empowered, and we're here to do the same for you. Don't miss out - [check it out now!](#)






Last Updated : 05 Jun, 2023

47

Similar Reads

	Logistic Regression Vs Random Forest Classifier		How to Solve Overfitting in Random Forest in Python Sklearn?
	How to Develop a Random Forest Ensemble in Python		Random Forest Hyperparameter Tuning in Python
	IBM HR Analytics on Employee Attrition & Performance using Rando...		Random Forest Classifier using Scikit-learn
	Hyperparameters of Random Forest Classifier		Differences between Random Forest and AdaBoost
	Calculate ROC AUC for Classification Algorithm Such as Random Forest		Random Forest for Time Series Forecasting using R

Related Tutorials

	OpenAI Python API - Complete Guide		Computer Vision Tutorial
	Pandas AI: The Generative AI Python Library		Top Computer Vision Projects (2023)
	Python for Kids - Fun Tutorial to Learn Python Programming		

Previous

Ensemble Methods in Python

Next

ML | Extra Tree Classifier for Feature Selection

Article Contributed By :

Avik_Dutta



Avik_Dutta

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Vote for difficulty

Current difficulty : [Easy](#)

Easy	Normal	Medium	Hard	Expert
------	--------	--------	------	--------

Improved By : [coderboy2901](#), [deepaymt](#), [devanshigupta1304](#), [gautamjain133](#), [rkbhola5](#), [abdulraheemshahzadbss2018a](#), [abhishekm482g](#), [parthshukla211](#)

Article Tags : [python](#) , [Machine Learning](#) , [Python](#)

Practice Tags : [Machine Learning](#), [python](#), [python](#)

[Improve Article](#)[Report Issue](#)

GeeksforGeeks
Simplified Education Private Limited
A-143, 9th Floor, Sovereign Corporate
Tower, Sector-136, Noida, Uttar Pradesh -
201305

feedback@geeksforgeeks.org



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

[About Us](#)[Legal](#)[Terms & Conditions](#)[Careers](#)[In Media](#)[Contact Us](#)[Advertise with us](#)[GFG Corporate Solution](#)[Placement Training Program](#)[Apply for Mentor](#)[Job-A-Thon Hiring Challenge](#)[Hack-A-Thon](#)[GfG Weekly Contest](#)[Offline Classes \(Delhi/NCR\)](#)[DSA in JAVA/C++](#)[Master System Design](#)[Master CP](#)[GeeksforGeeks Videos](#)

Languages

[Python](#)[Java](#)[C++](#)[PHP](#)[GoLang](#)[SQL](#)[R Language](#)[Android Tutorial](#)

DSA Concepts

[Data Structures](#)[Arrays](#)[Strings](#)[Linked List](#)[Algorithms](#)[Searching](#)[Sorting](#)[Mathematical](#)[Dynamic Programming](#)

DSA Roadmaps

[DSA for Beginners](#)[Basic DSA Coding Problems](#)[DSA Roadmap by Sandeep Jain](#)[DSA with JavaScript](#)[Top 100 DSA Interview Problems](#)[All Cheat Sheets](#)

Web Development

[HTML](#)[CSS](#)[JavaScript](#)[Bootstrap](#)[ReactJS](#)[AngularJS](#)[NodeJS](#)[Express.js](#)[Lodash](#)

[Operating Systems](#)[Django Tutorial](#)[Computer Network](#)[Python Projects](#)[Database Management System](#)[Python Tkinter](#)[Software Engineering](#)[OpenCV Python Tutorial](#)[Digital Logic Design](#)[Python Interview Question](#)[Engineering Maths](#)

Data Science & ML

DevOps

[Data Science With Python](#)[Git](#)[Data Science For Beginner](#)[AWS](#)[Machine Learning Tutorial](#)[Docker](#)[Maths For Machine Learning](#)[Kubernetes](#)[Pandas Tutorial](#)[Azure](#)[NumPy Tutorial](#)[GCP](#)[NLP Tutorial](#)[Deep Learning Tutorial](#)

Competitive Programming

System Design

[Top DSA for CP](#)[What is System Design](#)[Top 50 Tree Problems](#)[Monolithic and Distributed SD](#)[Top 50 Graph Problems](#)[Scalability in SD](#)[Top 50 Array Problems](#)[Databases in SD](#)[Top 50 String Problems](#)[High Level Design or HLD](#)[Top 50 DP Problems](#)[Low Level Design or LLD](#)[Top 15 Websites for CP](#)[Crack System Design Round](#)[System Design Interview Questions](#)

Interview Corner

GfG School

[Company Wise Preparation](#)[CBSE Notes for Class 8](#)[Preparation for SDE](#)[CBSE Notes for Class 9](#)[Experienced Interviews](#)[CBSE Notes for Class 10](#)[Internship Interviews](#)[CBSE Notes for Class 11](#)[Competitive Programming](#)[CBSE Notes for Class 12](#)

Commerce

Accountancy
Business Studies
Economics
Human Resource Management (HRM)
Management
Income Tax
Finance
Statistics for Economics

SSC/ BANKING

SSC CGL Syllabus
SBI PO Syllabus
SBI Clerk Syllabus
IBPS PO Syllabus
IBPS Clerk Syllabus
Aptitude Questions
SSC CGL Practice Papers

UPSC

Polity Notes
Geography Notes
History Notes
Science and Technology Notes
Economics Notes
Important Topics in Ethics
UPSC Previous Year Papers

Write & Earn

Write an Article
Improve an Article
Pick Topics to Write
Share your Experiences
Internships

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved