Random Forest Regression in Python

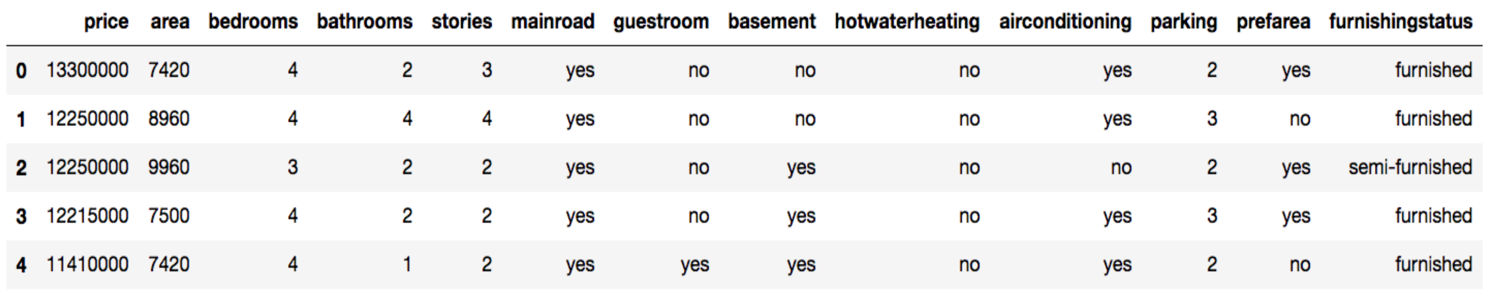
Random Forest:
The random forest algorithm, proposed by L. Breiman in 2001, has been extremely successful as a general-purpose classification and regression method. The approach, which combines several randomized decision trees and aggregates their predictions by averaging, has shown excellent performance in settings where the number of variables is much larger than the number of observations. Moreover, it is versatile enough to be applied to large-scale problems, is easily adapted.

Regression:
Given data on predictor variables (inputs, X) and a continuous response variable (output, Y) build a model for:
– Predicting the value of the response from the predictors.
– Understanding the relationship between the predictors and the response.
Example: predict a house price based on its area, bedrooms, bathrooms, stories, parking, etc.

To implement the random forest regressor, we're going to use scikit-learn, and we'll import our RandomForestRegressor from sklearn.ensemble.

```
#import required libraries and dataset
from sklearn.datasets import load_diabetes
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
import pandas as pd
df=pd.read_csv("Housing.csv")
df.head()
```

Output:



	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
0	13300000	7420	4	2	3	yes	no	no	no	yes	2	yes	furnished
1	12250000	8960	4	4	4	yes	no	no	no	yes	3	no	furnished
2	12250000	9960	3	2	2	yes	no	yes	no	no	2	yes	semi-furnished
3	12215000	7500	4	2	2	yes	no	yes	no	yes	3	yes	furnished
4	11410000	7420	4	1	2	yes	yes	yes	no	yes	2	no	furnished

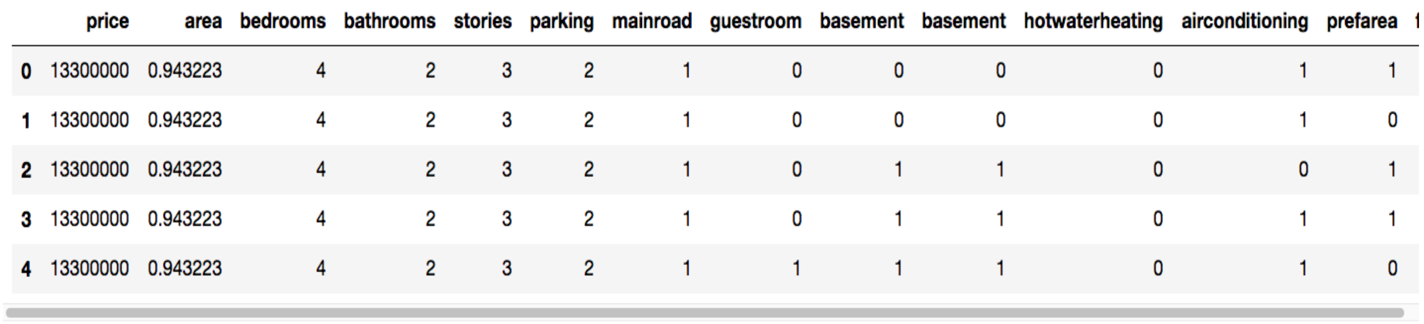
Data pre-processing:

We need to do some pre-processing as follows on dataset,use label encoder for label features like(furnishingstatus) for categorical variables as follows,

```
cat_df=df[["mainroad","guestroom","basement","basement","hotwat
erheating","airconditioning","prefarea","furnishingstatus"]]
num_df=df[["price","area","bedrooms","bathrooms","stories","par
king"]]
cat_df = cat_df.apply(LabelEncoder().fit_transform)
num_df["mainroad"]=cat_df["mainroad"]
new_df=pd.merge(num_df,cat_df,on="mainroad")
#use standard scaler to scale column value of area
ss=StandardScaler()
new_df[["area"]] =ss.fit_transform(new_df[["area"]])
new_df.head()
```

Output:

After encoding resultant data frame is as follows,



	price	area	bedrooms	bathrooms	stories	parking	mainroad	guestroom	basement	basement	hotwaterheating	airconditioning	prefarea
0	13300000	0.943223	4	2	3	2	1	0	0	0	0	1	1
1	13300000	0.943223	4	2	3	2	1	0	0	0	0	1	0
2	13300000	0.943223	4	2	3	2	1	0	1	1	0	0	1
3	13300000	0.943223	4	2	3	2	1	0	1	1	0	1	1
4	13300000	0.943223	4	2	3	2	1	1	1	1	0	1	0

Reshape data in required format:

Prepare dataset for machine learning algorithm readable format,

```
#separate dependent y and independent x variables
x=new_df.iloc[:,1:]
y=new_df.iloc[:,:1]
xtrain,xtest,ytrain,ytest=train_test_split(x,
y,test_size=0.2,random_state=32)
```

Train the model:

Apply RandomForestRegressor with parameter value of n_estimators and criterion ,100 and “mean square error(mse)” as follows,

```
#apply regression model
rfregressor=RandomForestRegressor(n_estimators=100,
criterion='mse')
rfregressor.fit(xtrain,ytrain)
```

```
ypred=rfregressor.predict(xtest)
```

Model Evaluation:
Check the accuracy of model as fallows,

```
#to check the accuracy of model
rfregressor.score(xtest,ytest)*100
```

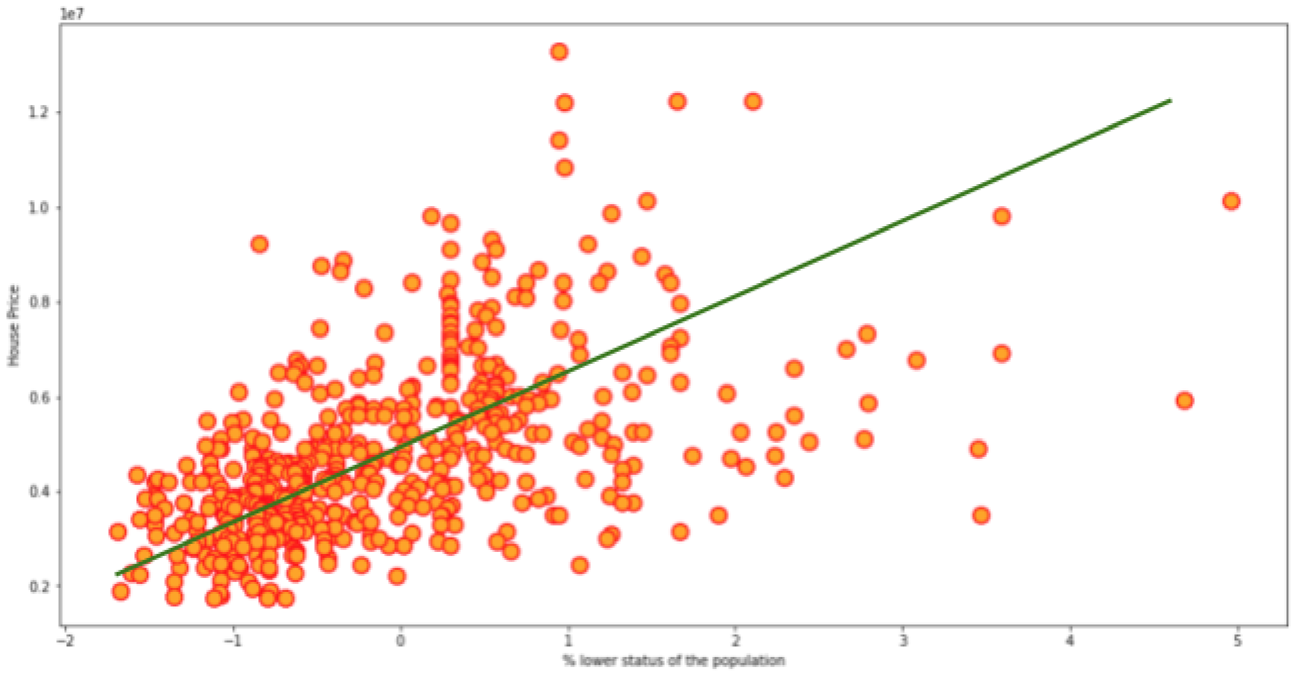
Output:

```
98.05266878151701
```

Plotting Result:
Plot result using scatterplot function of value status of population vs house price as follows,

```
# Plot the results
import matplotlib.pyplot as plt
plt.figure(figsize=(16, 8))
plt.scatter(x.iloc[:,:1],y, c='orange',edgecolor='red', s=120)
plt.plot(xtest.iloc[:,:1], ypred,color='green',lw=2)
plt.xlabel('% lower status of the population')
plt.ylabel('House Price')
plt.show()
```

Output:


1e7
House Price
% lower status of the population