

Untitled4

February 11, 2019

```
In [3]: import pandas as pd
import numpy as np
import matplotlib.pyplot as pyplot
import seaborn as sns
from sklearn.datasets import load_boston
from sklearn.metrics import mean_squared_error

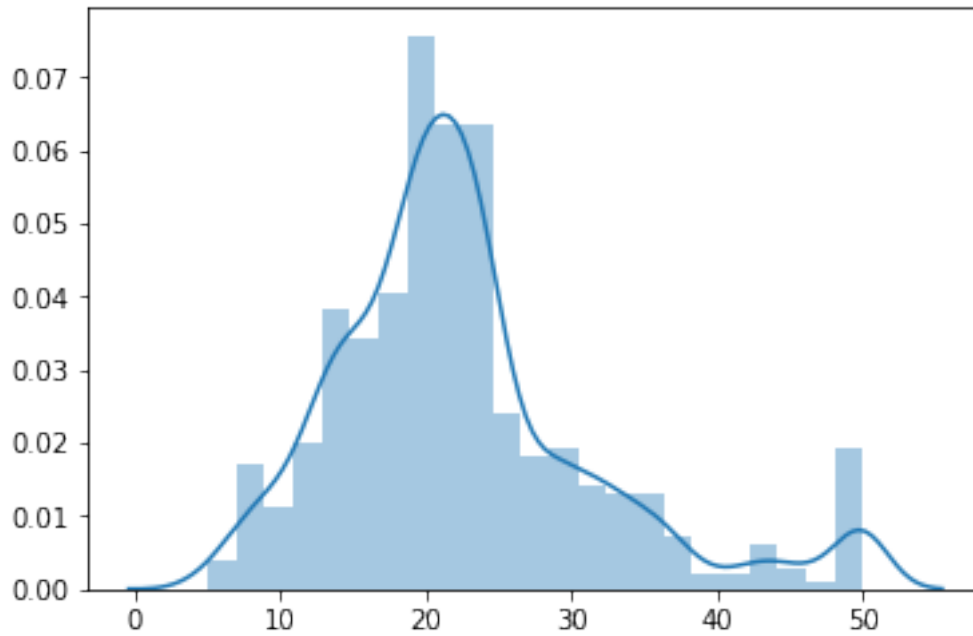
boston_data = load_boston()
#print(boston_data.DESCR)
X = boston_data.data
y = boston_data.target

X_df = pd.DataFrame(X, columns=boston_data.feature_names)
X_df.head(5)

len(y)

Out[3]: 506

In [4]: sns.distplot(y)
pyplot.show()
```

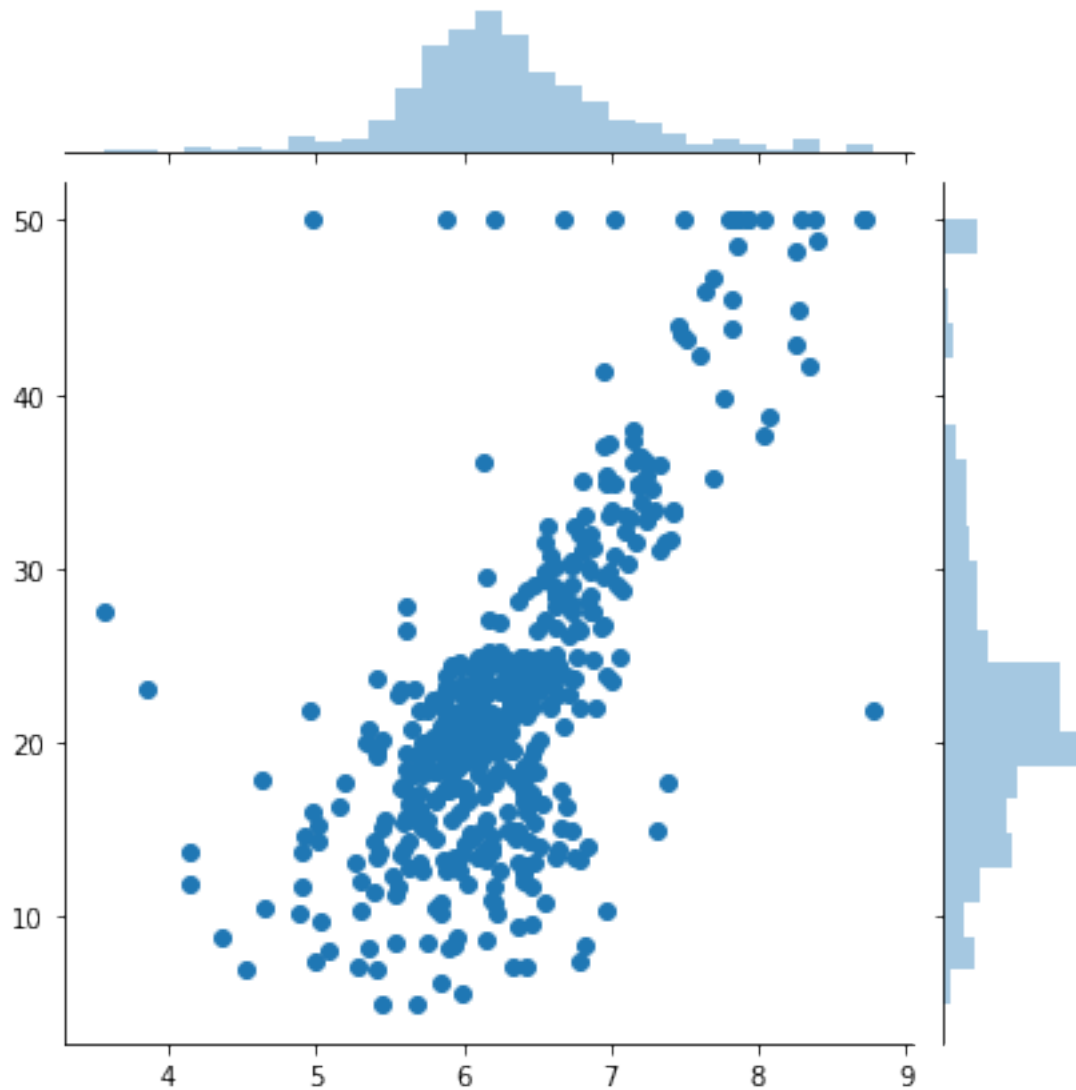


```
In [5]: y_pred = [0] * 506
        error = mean_squared_error(y, y_pred)
        print(error)

        sns.jointplot(X[:, 5], y)
```

592.146916996

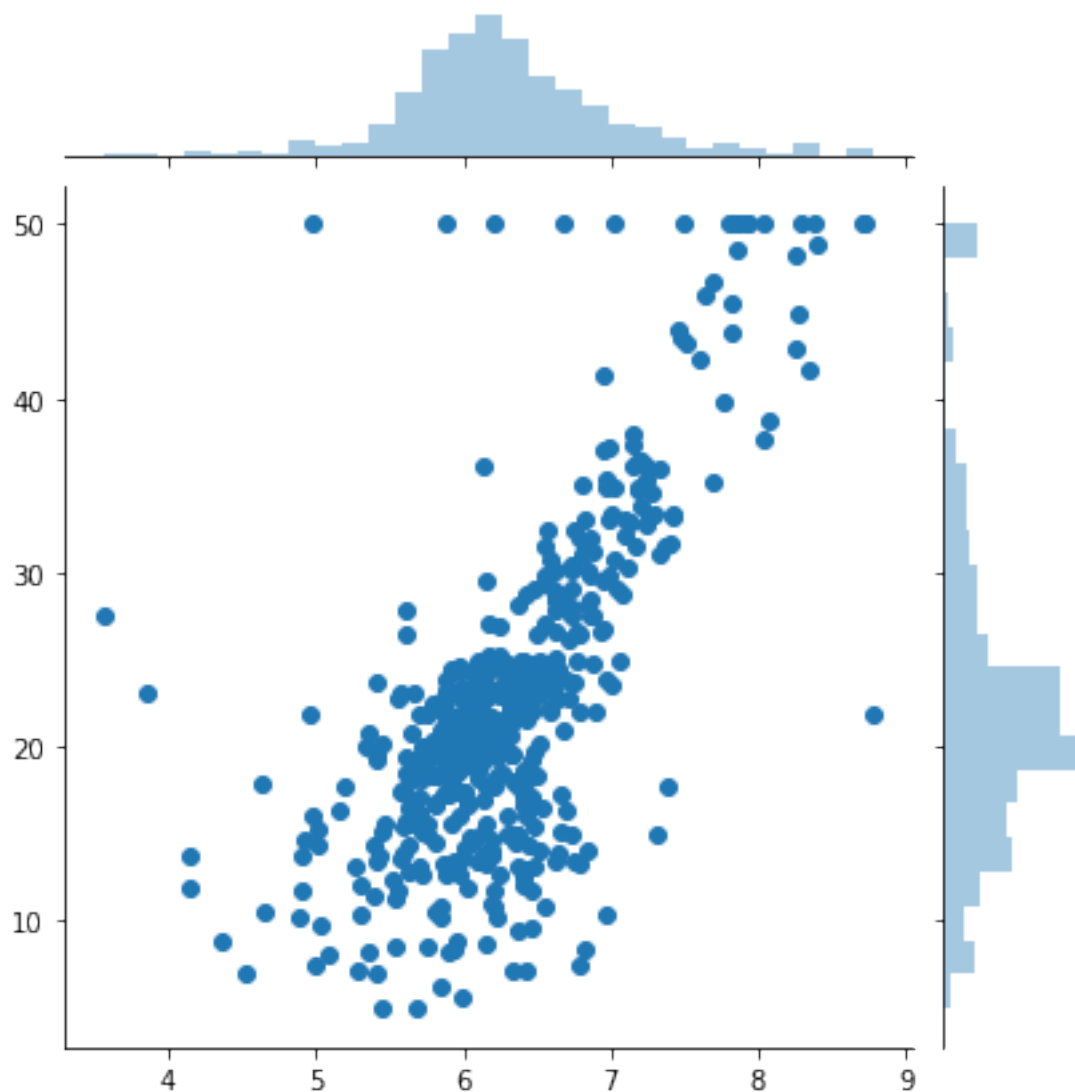
Out [5]: <seaborn.axisgrid.JointGrid at 0x7fe3268ada90>



```
In [6]: def manual_model(house):  
        return (house[5] - 4) * 10  
  
        y_pred = [manual_model(x) for x in X]  
        error = mean_squared_error(y, y_pred)  
        print(error)  
  
        sns.jointplot(X[:, 5], y)
```

44.0960725296

Out[6]: <seaborn.axisgrid.JointGrid at 0x7fe3267ca630>



```
In [7]: from sklearn.linear_model import LinearRegression
        from sklearn.model_selection import train_test_split

        model = LinearRegression()
        X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.33, random_state=42)
        model.fit(X_train, y_train)
        y_pred_ml = model.predict(X_test)
        error = mean_squared_error(y_test, y_pred_ml)
        print(error)

        print('-----')
        print('-----')
        print('-----')
```

```
model2 = LinearRegression(normalize=True)
model2.fit(X_train, y_train)
y_pred_ml_normalized = model2.predict(X_test)
error = mean_squared_error(y_test, y_pred_ml_normalized)
print(error)
```

20.7240234373

20.7240234373