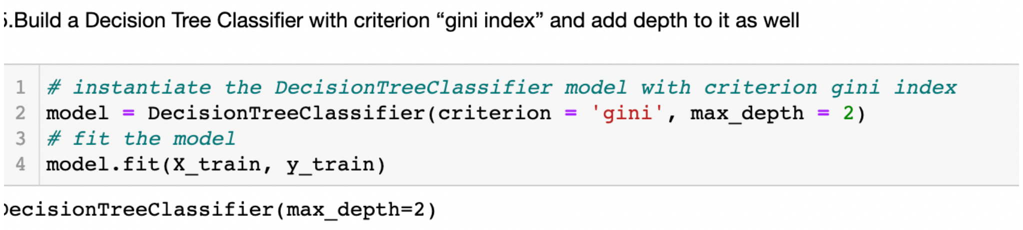
ICE-9

Ananya Samala-11527196

Decision tree and Linear Regression

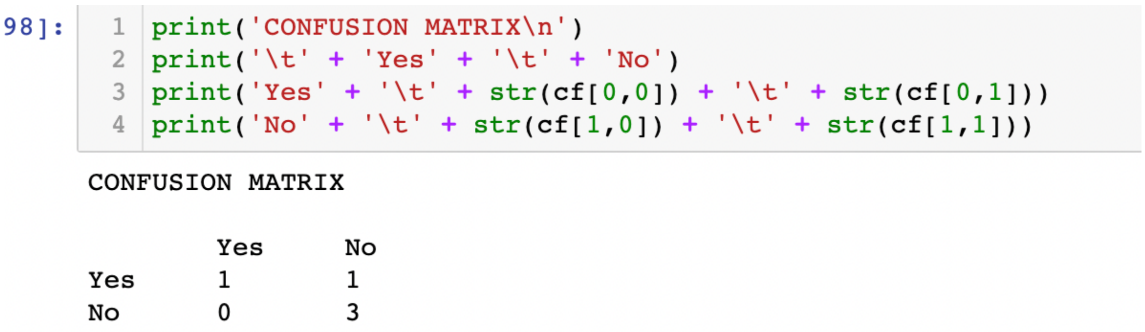
1. “Gini index”



5.Build a Decision Tree Classifier with criterion “gini index” and add depth to it as well

```
# instantiate the DecisionTreeClassifier model with criterion gini index
model = DecisionTreeClassifier(criterion = 'gini', max_depth = 2)
# fit the model
model.fit(X_train, y_train)
```

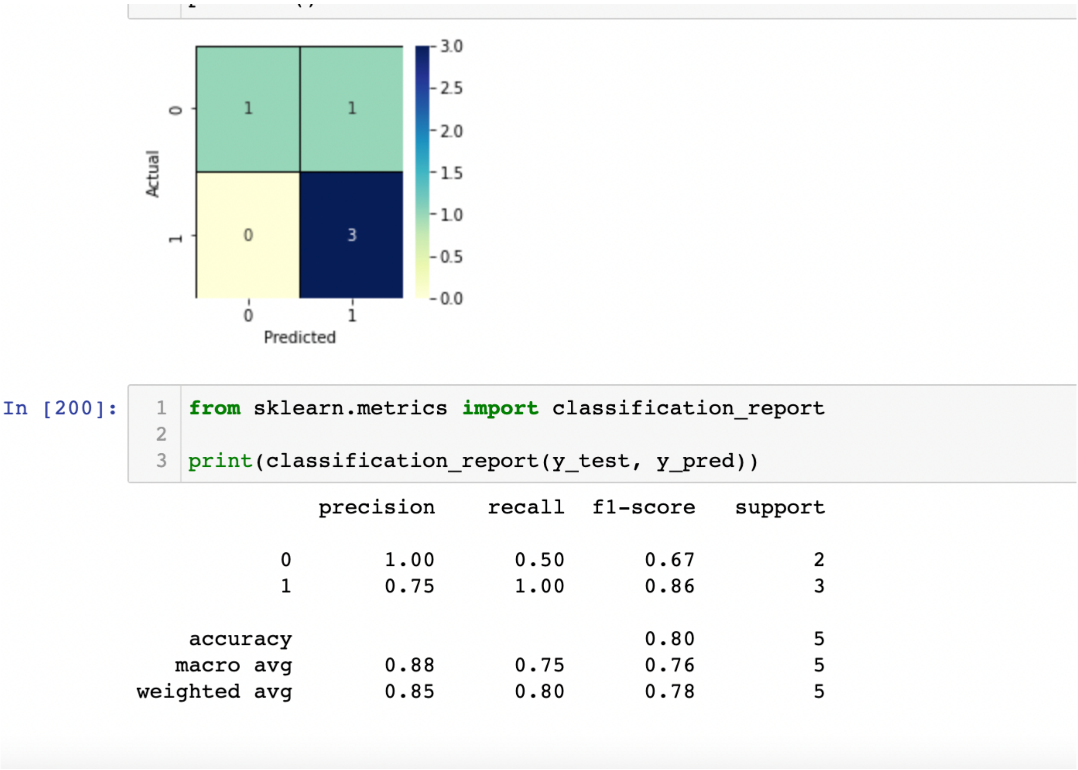
```
DecisionTreeClassifier(max_depth=2)
```



98]:

```
print('CONFUSION MATRIX\n')
print('\t' + 'Yes' + '\t' + 'No')
print('Yes' + '\t' + str(cf[0,0]) + '\t' + str(cf[0,1]))
print('No' + '\t' + str(cf[1,0]) + '\t' + str(cf[1,1]))
```

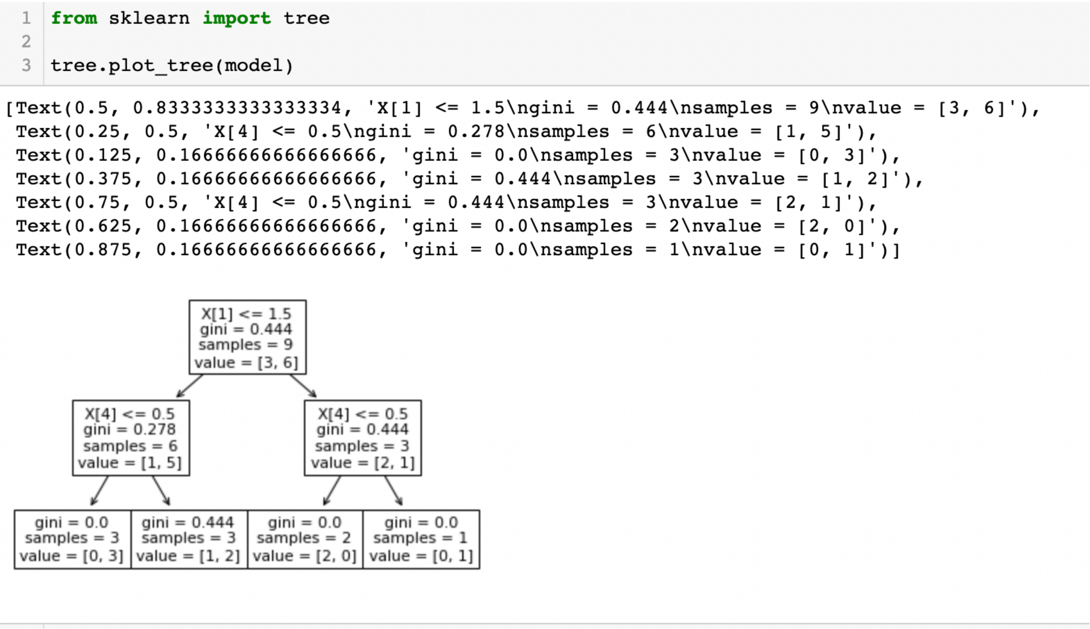
```
CONFUSION MATRIX

        Yes     No
Yes     1       1
No      0       3
```



In [200]:

```
from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred))
```

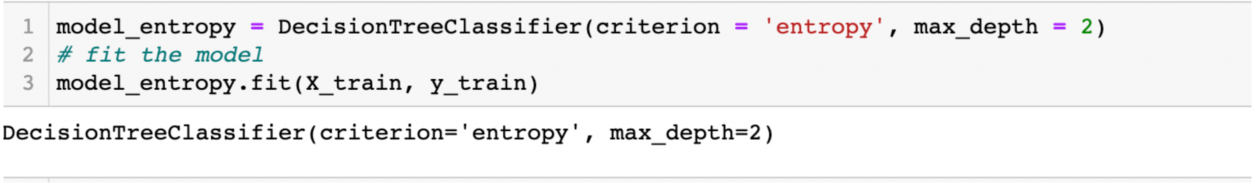
```
              precision    recall  f1-score   support

           0       1.00      0.50      0.67         2
           1       0.75      1.00      0.86         3

    accuracy                           0.80         5
   macro avg       0.88      0.75      0.76         5
weighted avg       0.85      0.80      0.78         5
```



```
from sklearn import tree

tree.plot_tree(model)
```

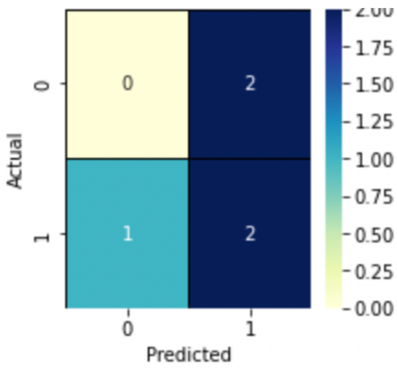
```
[Text(0.5, 0.8333333333333334, 'X[1] <= 1.5\ngini = 0.444\nsamples = 9\nvalue = [3, 6]'),
 Text(0.25, 0.5, 'X[4] <= 0.5\ngini = 0.278\nsamples = 6\nvalue = [1, 5]'),
 Text(0.125, 0.16666666666666666, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.375, 0.16666666666666666, 'gini = 0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(0.75, 0.5, 'X[4] <= 0.5\ngini = 0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.625, 0.16666666666666666, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.875, 0.16666666666666666, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]')]
```

2. “Entropy”

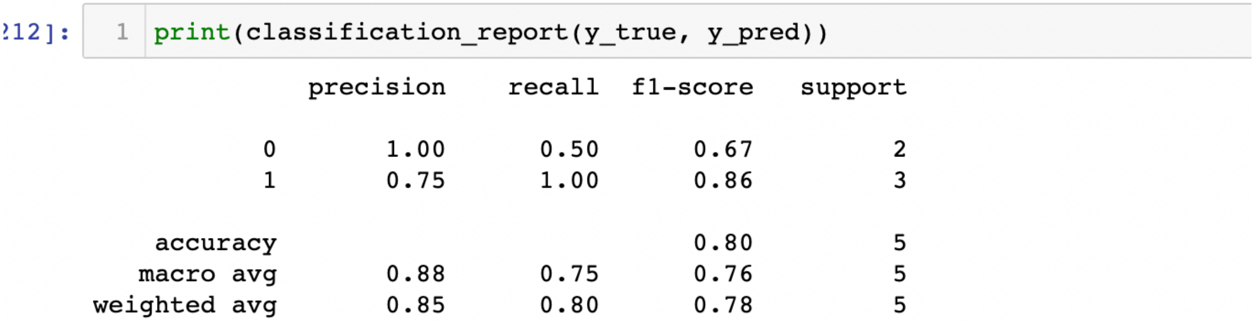


```python
model_entropy = DecisionTreeClassifier(criterion = 'entropy', max_depth = 2)
# fit the model
model_entropy.fit(X_train, y_train)
```

```
DecisionTreeClassifier(criterion='entropy', max_depth=2)
```

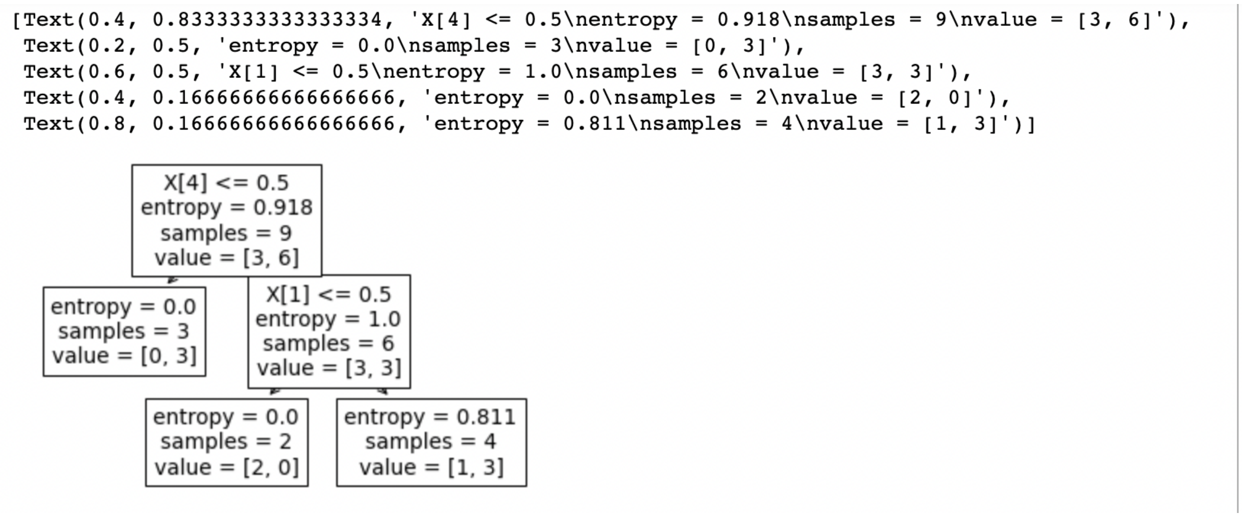


8. Explain whether the model is overfitting or underfitting



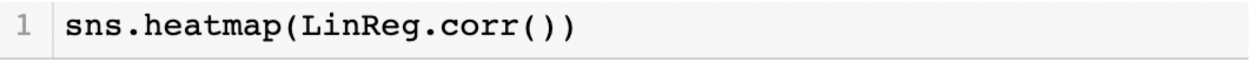
```python
print(classification_report(y_true, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      0.50      0.67         2
           1       0.75      1.00      0.86         3

    accuracy                           0.80         5
   macro avg       0.88      0.75      0.76         5
weighted avg       0.85      0.80      0.78         5
```



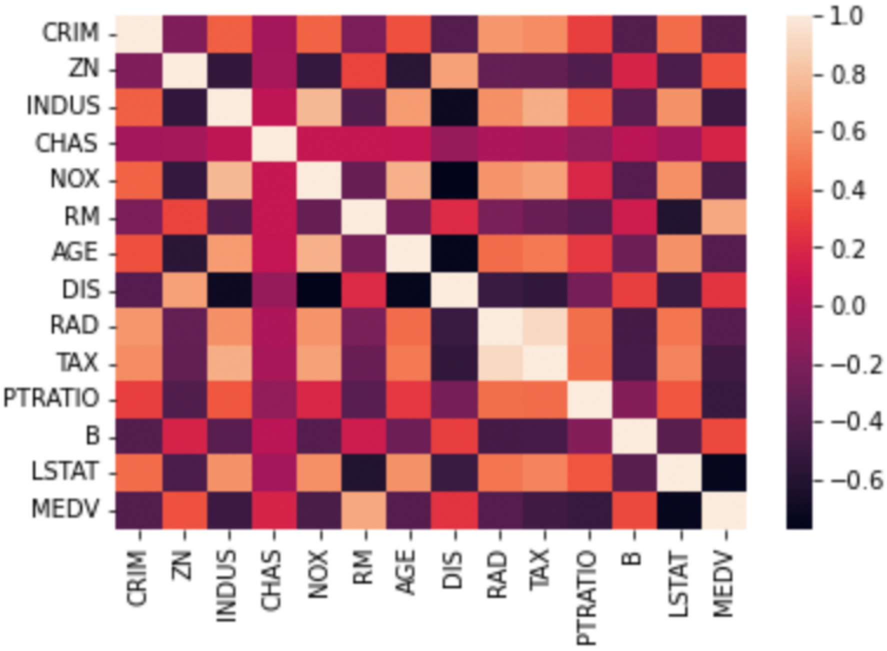
```
[Text(0.4, 0.8333333333333334, 'X[4] <= 0.5\nentropy = 0.918\nsamples = 9\nvalue = [3, 6]'),
 Text(0.2, 0.5, 'entropy = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.6, 0.5, 'X[1] <= 0.5\nentropy = 1.0\nsamples = 6\nvalue = [3, 3]'),
 Text(0.4, 0.16666666666666666, 'entropy = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.8, 0.16666666666666666, 'entropy = 0.811\nsamples = 4\nvalue = [1, 3]')]
```

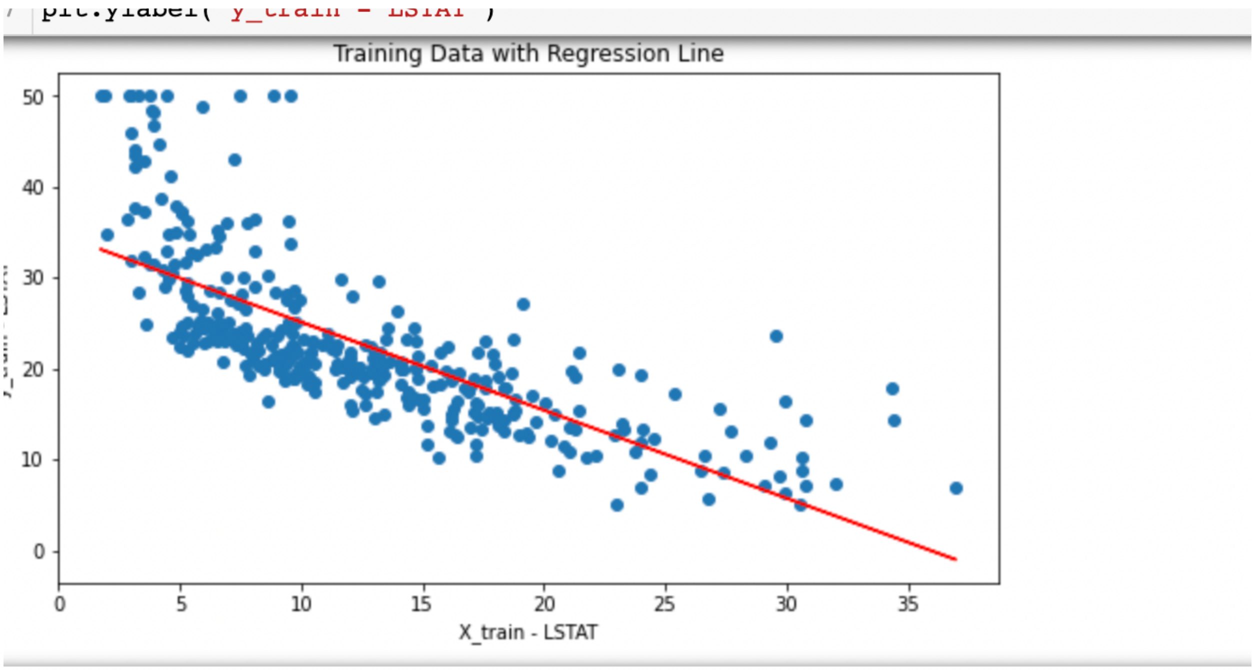
LINEAR REGRESSION



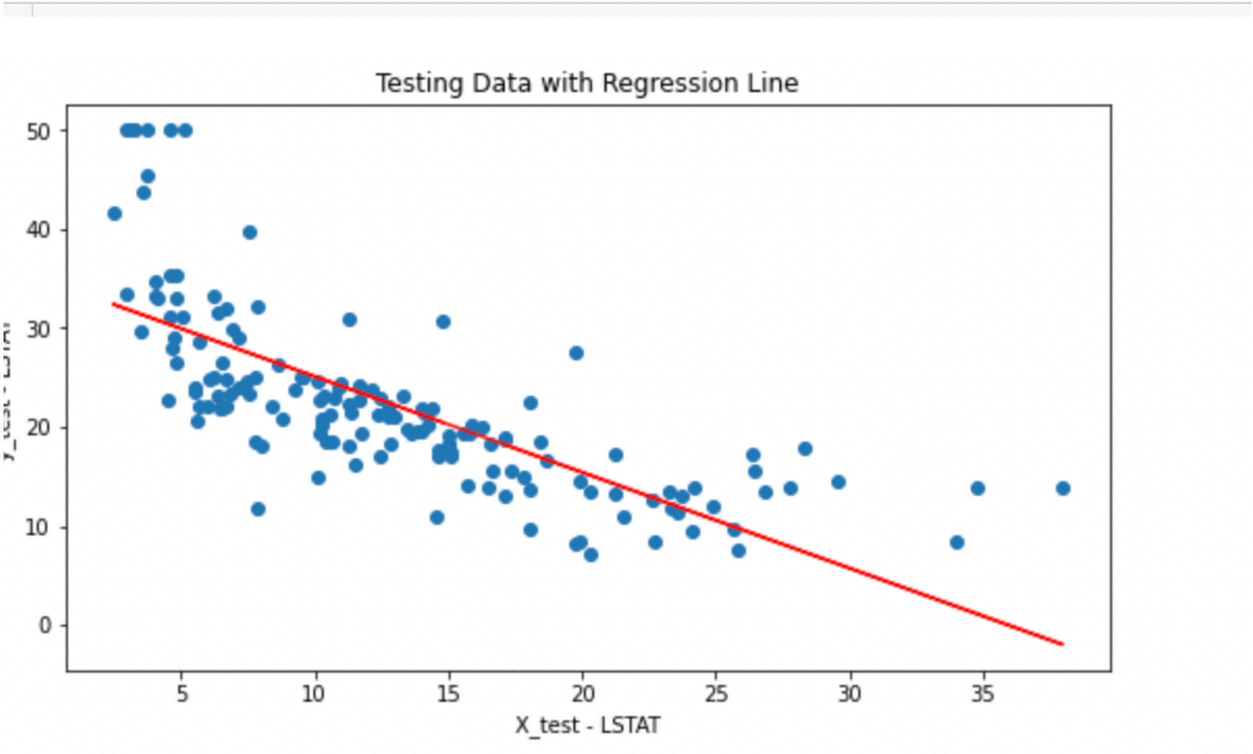
```
sns.heatmap(LinReg.corr())
```

```
<AxesSubplot:>
```



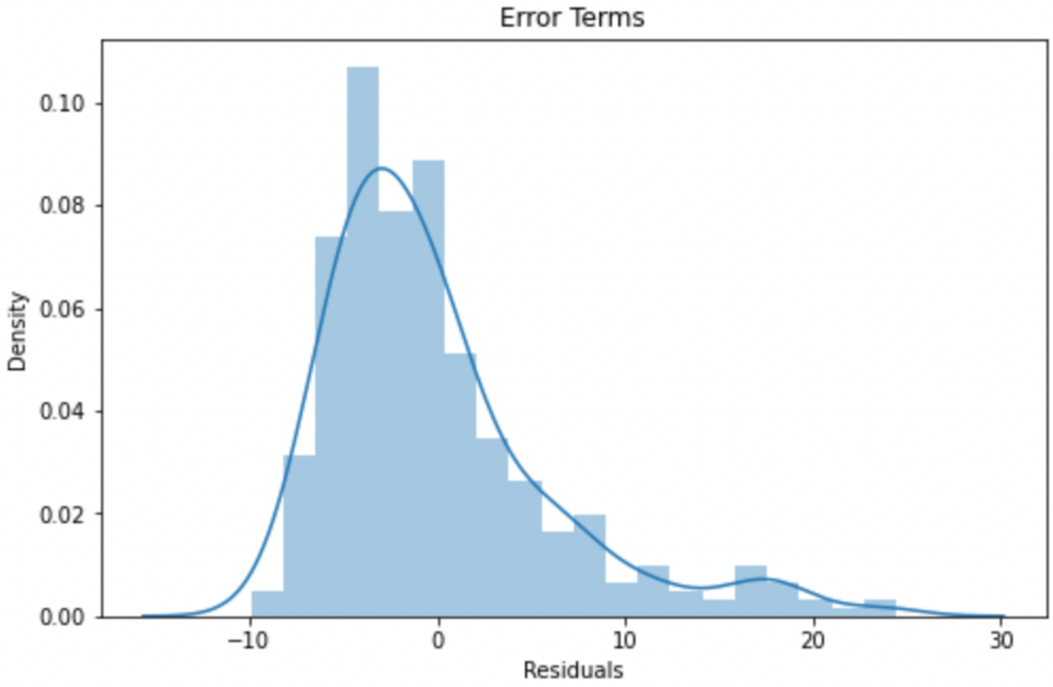


```
plt.ylabel('y_train - LSTAT')
```



```
  warnings.warn(msg, FutureWarning)
```

```
Text(0.5, 0, 'Residuals')
```



+++