1) Statistical Analysis and Data Exploration

a. Number of data points (houses)? 506

b. Number of features? 14

1. - CRIM per capita crime rate by town
2. - ZN proportion of residential land zoned for lots over 25,000 sq.ft.
3. - INDUS proportion of non-retail business acres per town
4. - CHAS Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
5. - NOX nitric oxides concentration (parts per 10 million)
6. - RM average number of rooms per dwelling
7. - AGE proportion of owner-occupied units built prior to 1940
8. - DIS weighted distances to five Boston employment centres
9. - RAD index of accessibility to radial highways
10. - TAX full-value property-tax rate per $10,000
11. - PTRATIO pupil-teacher ratio by town
12. - B 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
13. - LSTAT % lower status of the population
14. - MEDV Median value of owner-occupied homes in $1000's

c. Minimum and maximum housing prices?

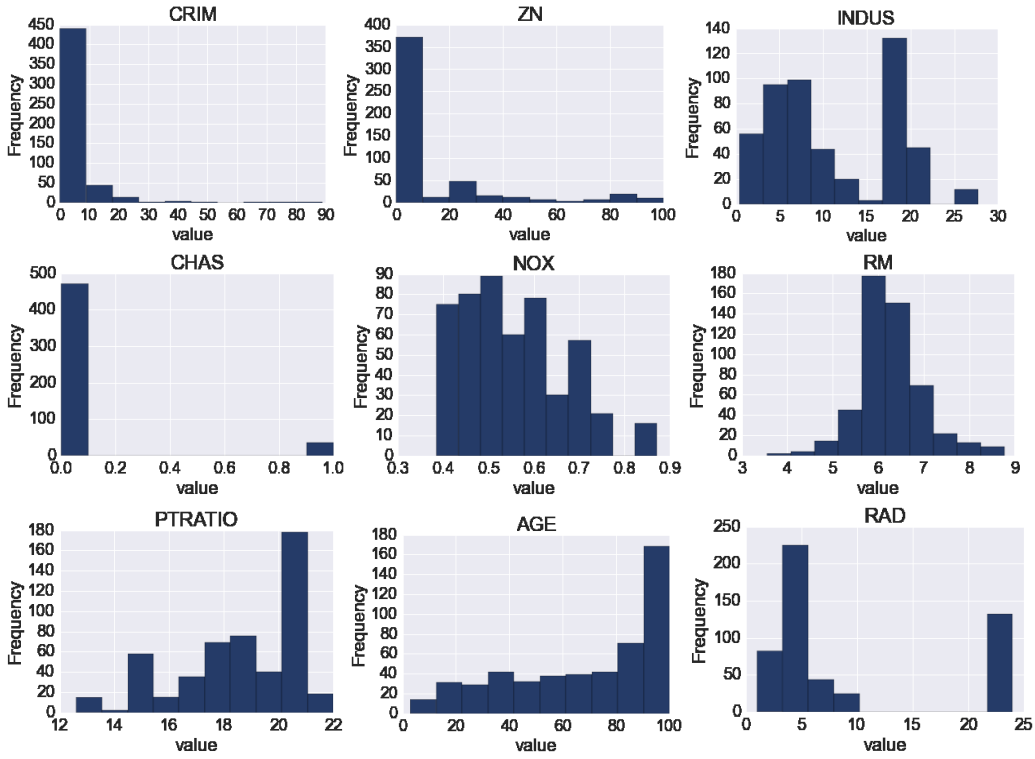
- Minimum value: 5.0
- Maximum value: 50.0

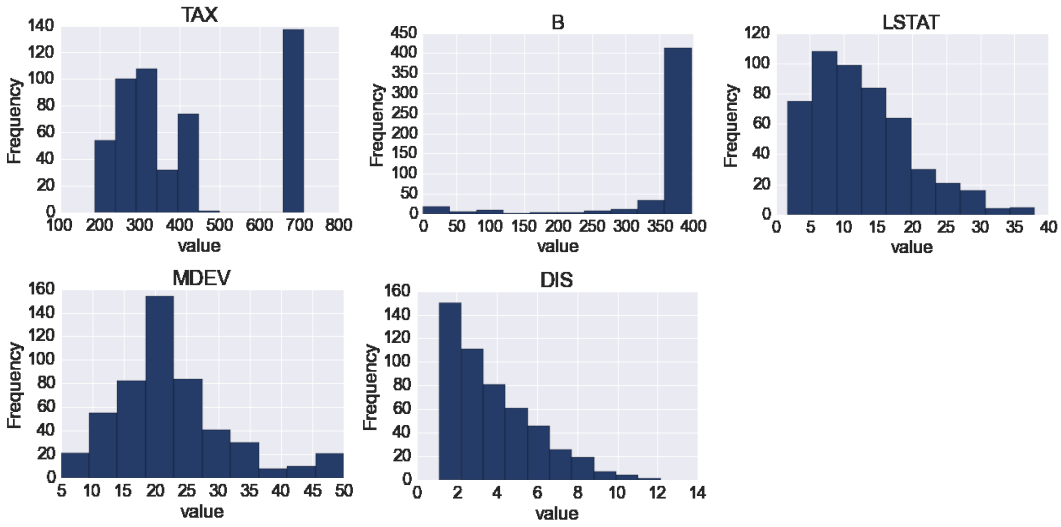
d. Mean and median Boston housing prices?

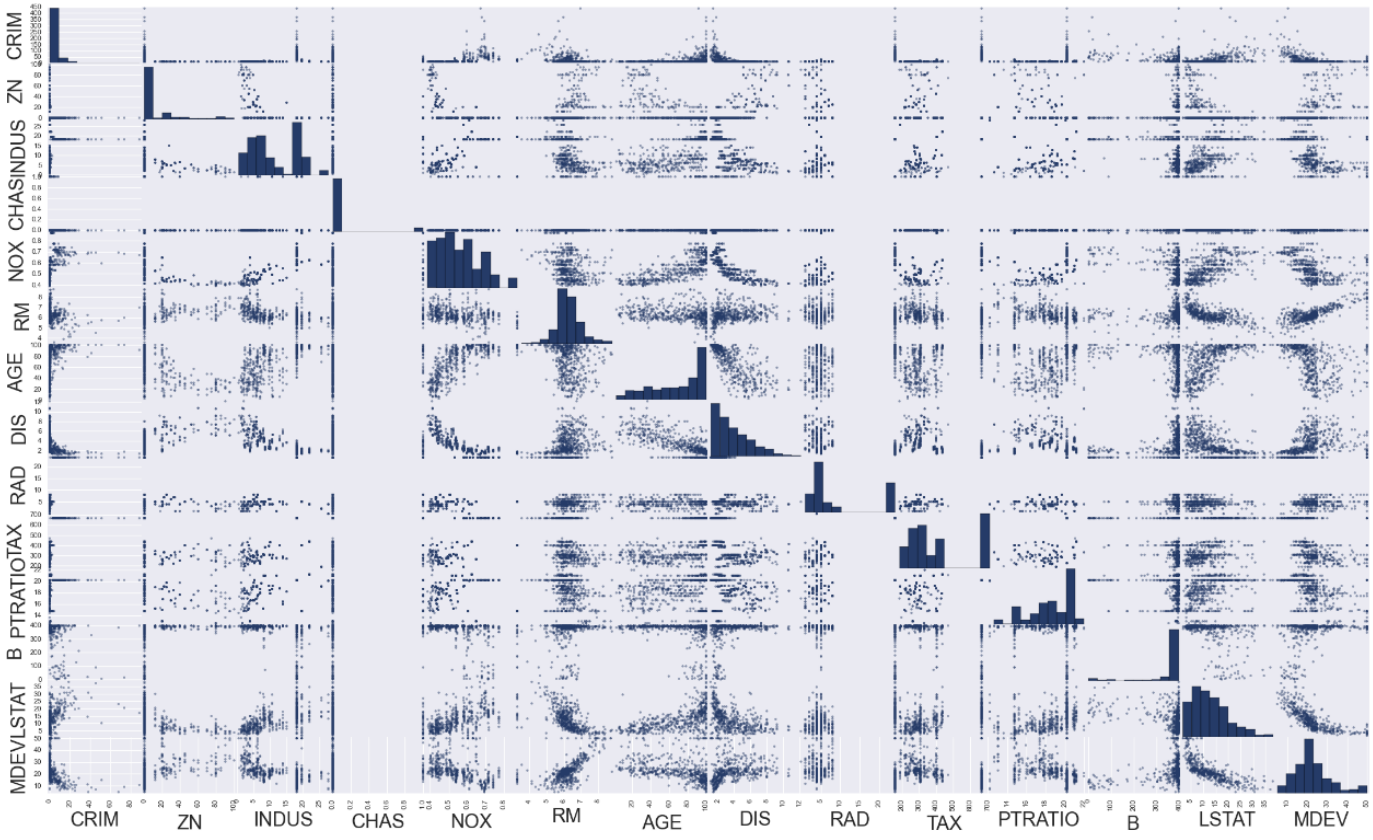
- Mean value: 22.5328063241
- Median value: 21.2

e. Standard deviation?

- Standard Deviation value: 9.18801154528


CRIM
ZN
INDUS
CHAS
NOX
RM
PTRATIO
AGE
RAD
Frequency
value


TAX
B
LSTAT
MDEV
DIS
Frequency
value


CRIM
ZN
INDUS
CHAS
NOX
RM
AGE
DIS
RAD
TAX
PTRATIO
B
LSTAT
MDEV

2) Evaluating Model Performance

- *Which measure of model performance is best to use for predicting Boston housing data and analyzing the errors? Why do you think this measurement most appropriate? Why might the other measurements not be appropriate here?*

  In this example, training time and run time are irrelevant so max accuracy was the only performance that was reviewed.

  My model uses explained_variance_score to evaluate the quality of the model. In choosing the best model, I considered the result of certain types of error. In my uneducated opinion, this model would be used to guide brokers in guessing the price that the house would likely sell for. This means that the total amount of error (positive or negative) would be the most accurate and therefore the most useful. In a live project I would rather to consult with an industry professional before making this kind of decision as I don't really know much about the housing market.

	Selling price is high	Selling price is low
Prediction is high	$$$	House not sold
Prediction is low	Earned less than potential	Money saved

  In the end, my decision was to prefer total accuracy as I lack the domain knowledge to know if guessing too low is better than guessing too high. It may be that it is better to sell a house for less than total potential rather than loose the sale altogether. This is best answered by a real estate agent or the team that will be actually using the model.

  It may be that the best answer is a range of prices and probabilities that would allow the agent to speculate on their own.

- *Why is it important to split the Boston housing data into training and testing data? What happens if you do not do this?*

The result of this exorcise is to predict the value of a single house. In order to do this you need to know if the model is good at predicting values. Separating the data into a training and testing set gives you the ability to see how accurate your model is at predicting the target variable.

- *What does grid search do and why might you want to use it?*

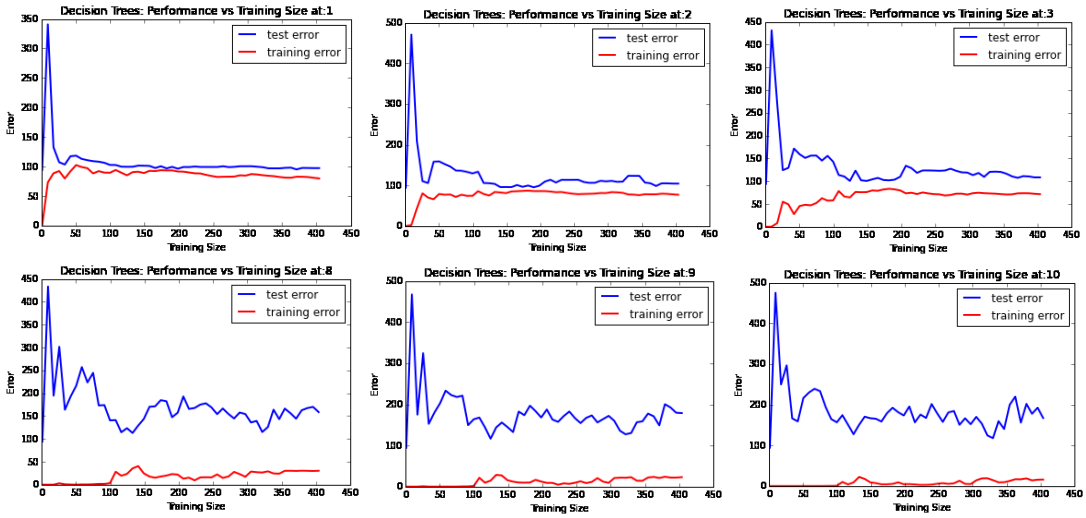
Grid search automates the process of cycling through a list of different model parameters (max_depth in this case) to choose which model parameters return the highest performance (accuracy in this case).

- *Why is cross validation useful and why might we use it with grid search?*

Cross validation enables us to us all of the data in the dataset to both train and test by iterating the testing sample over k bins. This prevents the training data from being unbalanced by outliers that are not in the test data. The average performance of all k groups is the overall performance of the model.

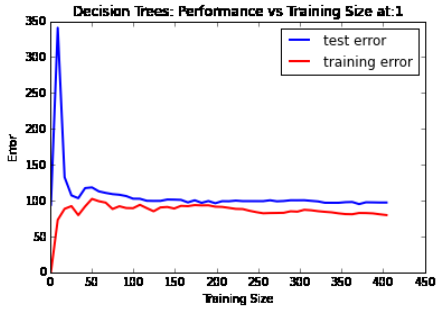
3) Analyzing Model Performance

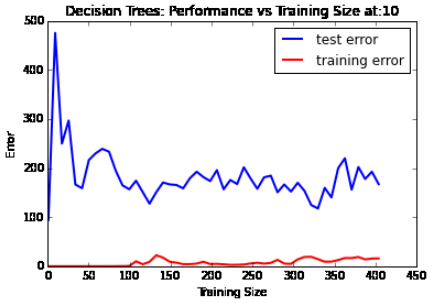
- *Look at all learning curve graphs provided. What is the general trend of training and testing error as training size increases?*



As the level of depth increases, the training accuracy improves while the test accuracy gets worse. At lower depth the training error is high as the prediction is made on a very small amount of information. At higher depth the model is tends to more closely fit the training data while making more error in the test data.

- *Look at the learning curves for the decision tree regressor with max depth 1 and 10 (first and last learning curve graphs). When the model is fully trained does it suffer from either high bias/underfitting or high variance/overfitting?*





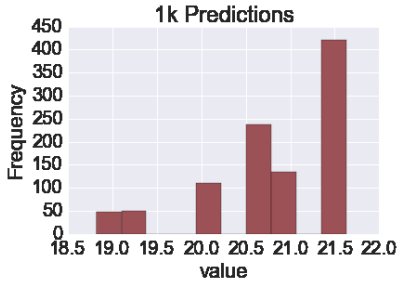
At MD=1 the model is underfitting as the test error and training error are almost the same. This is because the model is only using a small amount of information to predict. At MD=10 the model is overfitting as the training error is almost null while the testing error is much higher.

- *Look at the model complexity graph. How do the training and test error relate to increasing model complexity? Based on this relationship, which model (max depth) best generalizes the dataset and why?*

Test error and training error are inversely related. As the depth increases the model will fit the training data more closely but have more error predicting new values.

Model Prediction

- Model makes predicted housing price with detailed model parameters (max depth) reported using grid search. Note due to the small randomization of the code it is recommended to run the program several times to identify the most common/reasonable price/model complexity.



I set the **fit_predict_model(city_data)** to run a thousand times to see how distributed the guess is. You can see that the model will predict around 21.5 almost half of the time. All of the 1k guesses fell between 19 and 22. This would indicate that the model is fairly consistent with its guess.

- Compare prediction to earlier statistics and make a case if you think it is a valid model.

  It's within the min and max of the dataset. The range in the histogram is a lot lower than the standard deviation of the dataset. I decided to take a closer look at similar houses (houses that also sold in that price range:

```
df[(df['MDEV'] > 20.5) & (df['MDEV'] < 22.5)].mean()
df[(df['MDEV'] > 20.5) & (df['MDEV'] < 22.5)].corr()
```

	New Data	Mean of similar houses	Difference	Correlation to MDEV
CRIM	11.95	1.212243	10%	16%
ZN	0	9.377049		1%
INDUS	18.1	10.677213	59%	8%
CHAS	0	0.114754		11%
NOX	0.659	0.523262	79%	6%
RM	5.609	6.157459	110%	26%
AGE	90	59.496721	66%	8%
DIS	1.385	4.293872	310%	1%
RAD	24	7.688525	32%	17%
TAX	680	365.344262	54%	17%
PTRATIO	20.2	18.722951	93%	32%
B	332.09	386.460164	116%	3%
LSTAT	12.13	10.900656	90%	5%

  The output house seems to fit with 'similar' houses. This would indicate that the model is defendable.

I would probably try running the model again using only the variables that are strongly related to the target variable as opposed of including all variables and see how that changes the prediction. It may also be worth examining removing outlier houses to see if that improves the model prediction. In this problem, the dataset is assumed to be true and valid.

The standard deviation for the prediction is 0.8 (over 1k iterations). That is $800 or $404,800 total error potential over 506 houses. It would be up to the client to decide how accurate the prediction would need to be in order to decide if the model is worth using. Only if the model is more accurate than the 'best guess' of the experienced sales representatives then the model should be used.