Question 1

As a reminder, you can view a description of the Boston Housing dataset here, where you can find the different features under **Attribute Information**. The `MEDV` attribute relates to the values stored in our `housing_prices` variable, so we do not consider that a feature of the data.

Of the features available for each data point, choose three that you feel are significant and give a brief description for each of what they measure.

Remember, you can **double click the text box below** to add your answer!

Answer:

1) "DIS: weighted distances to five Boston employment centres" This gives the distances to where people would work. The number one rule in real estate is "location, location, location"

2) "PTRATIO: pupil-teacher ratio by town" The pupil-teacher ratio is schools is the closest thing I could find to a school rating. In the United States, the quality of the schools normally will drive the housing prices

3) "CRIM: per capita crime rate by town" The crime rate in the area might also be a good negative predictor of housing prices. Where the crime rate is high, I would expect lower home valuations, and where it's low higher valuations.

Question 2

Using your client's feature set `CLIENT_FEATURES`, which values correspond with the features you've chosen above?

Hint: Run the code block below to see the client's data.

```
[[11.95, 0.0, 18.1, 0, 0.659, 5.609, 90.0, 1.385, 24, 680.0, 20.2, 332.09, 12.13]]
```

Answer: CRIM = 11.95, DIS = 1.385, PTRATIO = 20.2

Question 3

Why do we split the data into training and testing subsets for our model?

Answer: We split the data into training and testing subsets for our model because it allows us to use the test set to estimate how well our model will generalize if it is given new data in the future. If the test set data is used for training, then our model will likely fit the data better. However, to get this better fit we run the risk of variance error in our model because the model may be fitting the current data very well, but not new data we will encounter.

Question 4

Which performance metric below did you find was most appropriate for predicting housing prices and analyzing the total error. Why?

- *Accuracy*
- *Precision*
- *Recall*
- *F1 Score*
- *Mean Squared Error (MSE)*
- *Mean Absolute Error (MAE)*

Answer: I felt that the mean squared error would be most appropriate performance metric for this project. The first reason for this is that the y data is continuous, which makes this a regression problem; Accuracy, precision, recall and F1 Score are more appropriate for categorical output data. That leaves MSE and MAE. I chose MSE over MAE because the squaring of errors in MSE penalizes predictions farther from the true value more than MAE. Although it is more susceptible to outliers, it should fit the data better.

Question 5

What is the grid search algorithm and when is it applicable?

Answer: The grid search algorithm is an optimization routine. For categorical and discrete values, it can be used to exhaustively explore a space to find the optimum solution given a set of parameters to vary and a specific performance metric. It can also be used for continuous data, but because it only tests the output (and calculates errors in this case) at the data points given, it will likely not find a local/global optimum when performing its search.

Question 6

What is cross-validation, and how is it performed on a model? Why would cross-validation be helpful when using grid search?

Answer: cross-validation takes a dataset and iteratively (1 through k) reserves a portion of the data to be used as a test set for error calculations. The non-reserved data in each iteration is used for training. Generally this is done in "folds" where all of the data is assigned to be part of one of the 'k' folds. The error in the test set from each iteration can then be used to calculate an average/mean cross-validation error. These mean errors can be compared to each other when doing a grid search to find the parameters that minimize the generalization error of the model. There are a couple advantages to using cross-validation over a standard train/test data split. One is that we are able to use as much of the data as possible to train our model. This benefit comes from iterating over the k-folds allowing all of the folds to be considered training and test data during different iterations. The

other benefit of cross-validation comes from avoiding overfitting. If we only use one training and test set, then there is a possibility that the random training set we choose may somehow be skewed. Running cross-validation helps to solve this problem with the k-fold methodology giving us a more representative understanding the data and a model that is more likely to generalize well.

Question 7

Choose one of the learning curve graphs that are created above. What is the max depth for the chosen model? As the size of the training set increases, what happens to the training error? What happens to the testing error?

Answer: I chose the graph in the lower left with a max_depth = 6. As the size of the training set increases, the training error increases slightly. However, the testing error decreases until roughly 250 data points and the appear to just noisily bounce around.

Question 8

Look at the learning curve graphs for the model with a max depth of 1 and a max depth of 10. When the model is using the full training set, does it suffer from high bias or high variance when the max depth is 1? What about when the max depth is 10?

Answer: For the model with max_depth = 1, the model is definitely suffering from high bias. This is obvious because the training and testing error are very high even at points on the graph with lots of data points (e.g. 150+). When the max_depth = 10, there appears to be high variance because the training error is very small even though there is training error. This suggests over-fitting when max_depth = 10.

Question 9

From the model complexity graph above, describe the training and testing errors as the max depth increases. Based on your interpretation of the graph, which max depth results in a model that best generalizes the dataset? Why?

Answer: In the model complexity graph above, the training error monotonically decreases as maximum depth increases. This is because the model is allowed to be more complex in order to fit the data it is given. The testing error decreases from maximum depth of 1 to 3, but then seems to hit a noisy plateau around 4. I would say that 4 is the ideal maximum depth in this model. I say that because models that use maximum depth over 5 will start to see variance error and not generalize well. I can see this from the chart because the training error is still decreasing at 5 and beyond, but the testing error no longer improves.

Question 10

Using grid search on the entire dataset, what is the optimal `max_depth` *parameter for your model? How does this result compare to your intial intuition?*

Hint: Run the code block below to see the max depth produced by your optimized model.

```
print "Final model optimal parameters:", reg.best_params_
```

```
Final model optimal parameters: {'max_depth': 4}
```

Answer: I re-ran the code 5 times getting new randomized test and training sets. The max depths I got from the optimal model parameters were 4,4,6,4,4. So the median is a max depth of 4. This result matches up exactly with the intiution I gained from looking at the learning curves and complexity model.

Question 11

With your parameter-tuned model, what is the best selling price for your client's home? How does this selling price compare to the basic statistics you calculated on the dataset?

Hint: Run the code block below to have your parameter-tuned model make a prediction on the client's home.

```
sale_price = reg.predict(CLIENT_FEATURES)

print "Predicted value of client's home: {0:.3f}".format(sale_price[0])
```

```
Predicted value of client's home: 21.630
```

Answer: The best selling price for my client's home is 21,630 dollars (remembering that the MEDV, median value, response variable is in 1,000's). This house price seems to be about "average". It is lower than the mean of 22,533 dollars, but higher than the median of 21,200 dollars.

Question 12 (Final Question):

In a few sentences, discuss whether you would use this model or not to predict the selling price of future clients' homes in the Greater Boston area.

Answer: While the process we used seems to have optimized the model to the data we have, I would not use this model to predict home prices is Boston. I would not want to use this model because I feel that it is missing key features that would be better suited at predicting home prices. Some examples of features that I feel are missing are square footage, school ratings for the home(e.g. Great Schools ratings) and age (the current data set tells us the proportion built prior to 1940, but not the numerical age of the homes). The other reason I would not want to use this model is that it is based on an aggregated data set. The response values are "Median values of owner-

occupied homes" so we're not even using values for individual homes, but a summary statistic to create our predictor.