Project-1 Report

1) Statistical Analysis and Data Exploration
    - Number of data points = 506
    - Number of features = 13
    - Minimum house price = 5.0; Maximum house price = 50.0
    - Mean of prices = 22.53; Median of prices = 21.2
    - Standard deviation of prices = 9.19

2) Evaluating Model Performance

    - Best measure of model performance:

      For regression models the best measures of performance are mean absolute error or mean squared error. These measurements are appropriate as they represent the collective difference between the original and predicted values and this difference or error value can be used to fine tune the model

      I would choose **mean squared error** as the measure of model performance as more emphasis has to be given to larger errors rather than smaller errors while evaluating the performance in this scenario.

      The other popular measures are accuracy, precision, recall, F1 score which give information about how properly model is classified but not about the prediction. To be short accuracy, precision, recall and F1 would work properly when predicting targets that are in discrete form but not targets that are in continuous form.

    - Importance of splitting data:

      Splitting the available data into two sets helps us to train the data on one set and then measure the performance of the trained model on the second set. This gives us the measure of how well the model has performed when exposed to data that it has never seen before. If we don't split the data, we can never know how good the model is predicting on new data and so we cannot validate if the trained model is performing well or not.

    - Grid Search:

      Every model has different sets of parameters that can be configured for training the model. For example, in SVM classifier we can use different kernel functions. Fine tuning these parameters by minimizing the error would increase the models performance. Using grid search we can provide all the sets of parameters we want the model to use for training and get the optimized model configured with best set of parameters from the set we have provided.

- Cross Validation

  Cross Validation is fine tuning the hyper parameters before testing the model on test set. For this purpose, the training set is divided into 'n' number of folds and model is trained and tested between these folds to obtain some metrics using which the best model is determined.
  Grid search allows us to do the cross validation automatically once we provide all the hyper parameters that should be considered for fine tuning the model.

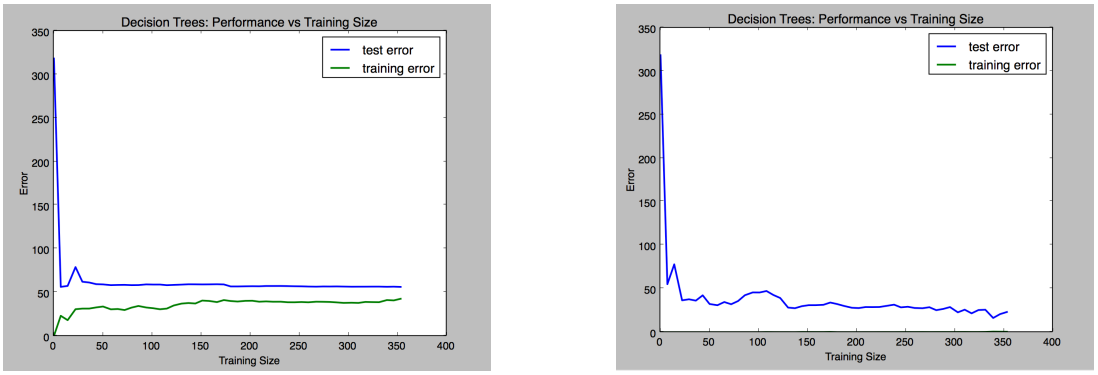
3) Analyzing Model Performance

- As the training size increases the training error gradually increases and the testing error gradually decreases until they both reach a plateau. And the difference between training and testing error reduces until it reaches a plateau.

- If a learning curve shows high error on both training and testing data, then the model is suffering from high bias and similarly if the learning curve shows large gap between training and testing sets then the model is suffering from high variance.

  Figure 1 shows the Learning curve graphs for models with max depth of 1 and 10. From (a) we can observe that the test error and training error are moving towards each other and finally settled with very less difference between them, so the model with max depth 1 has high bias.

  From (b) we can observe that the learning curve for the model with max depth 10, the magnitude of training and testing errors are low but the difference between them is very high, so this model has high variance.



(a) (b)

Figure 1: Max depth 1 and 10 Learning curve graphs; (a) : Graph with depth 1 ; (b) : Graph with depth 10

- We can observe that the graph shows how the model performs as the its model complexity increases. Several observations can be made, here are two if them

If we compare the training and the testing curves individually, as the model complexity is increasing the training error has decreased eventually reaching a plateau and the testing error has almost stabilized at a certain magnitude.

Here is comparison if we compared the training and testing error together. Initially when the model is less complex the difference between the training and test errors is low but the magnitudes of the errors itself is very high, this explains the high bias and low variance condition of the model when its complexity is low.

If we compare the training and test errors when the complexity is high then we can observe that magnitudes of both the errors are far apart from each other, which explains the high variance and low bias condition of the model.

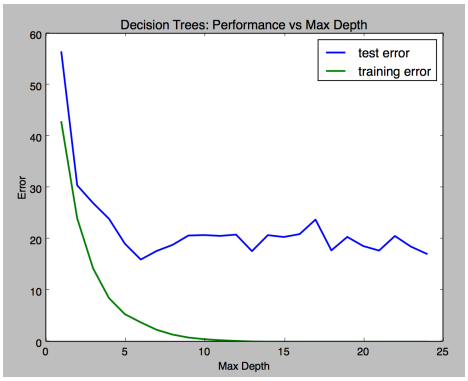


Figure 2: Performance vs Max Depth graph

- At the depth of 6 the training error is 3.77 and testing error: 16.00 with the difference between testing and training error of 12.24. The model with depth of 6 is the optimal model as it has best difference between training and test error while considering both their magnitudes (That is both the magnitudes seem to be considerably low when compared with other depths). And because of this reason at depth for 6 the model has proper trade off between bias and varience.

4) Model Prediction
    - The predicted price of the house is 19.93
    - The predicted price is -0.28 standard deviations away from the mean so it seems reasonable enough. (Didn't understand the question properly.)

References:

1) http://scikit-learn.org/stable/documentation.html
2) http://datascience.stackexchange.com/questions/361/when-is-a-model-underfitted
3) Udacity Machine Learning Nano Degree Course content
4) Udacity Introduction to Machine Learning by Sabastian Thrun
   https://www.udacity.com/course/intro-to-machine-learning--ud120