**Does Not Meet Specifications**

**Quality of Code**

**Specification**

All coding sections run fine without error. Students do not alter the starting code beyond what is requested.

Meets Specification

**Data Exploration**

**Specification**

All requested information about the Boston housing data is accurately supplied to within a tolerance of +/-0.05. Students use the numpy library in the provided template code to obtain results.

Does Not Meet Specification

**Reviewer Comments**

The number of data points and the number of features are not correct, the number of data points is the number of cases or houses, basically the number of rows. The number of features is the number of characteristics, attributes of each data point.

**Evaluating Model Performance**

**Specification**

Student uses an appropriate error scoring metric from the list provided and provides ample reasons for doing so.

Meets Specification

**Specification**

Student provides a valid reason why data is split into training and testing data. Student implements this split in code.

Does Not Meet Specification

**Reviewer Comments**

I'm not fully able to understand why train test split is performed by using cross validation:

kf =cross\_validation.KFold(len(city\_data.data),3, shuffle= True, random\_state=0)

This way your producing three folds and using just one of those and you don't achieve the mandatory requirement to split the data into 70% training and 30% testing.

I would suggest you to use:

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.30, random\_state=42)

<http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.train_test_split.html>

**Specification**

Student explains the importance of cross validation and why it is useful for grid search to use. If a student modifies grid search beyond the default 3-fold cross validation a reasonable justification is provided.

Meets Specification

**Specification**

Student properly implements grid search and justifies why.

Meets Specification

**Reviewer Comments**

**Pro tip:** There are other techniques that could be used for hyperparameter optimization in order to save time like [RandomizedSearchCV](http://scikit-learn.org/stable/modules/generated/sklearn.grid_search.RandomizedSearchCV.html#sklearn.grid_search.RandomizedSearchCV), in this case instead of exploring the whole parameter space just a fixed number of parameter settings is sampled from the specified distributions. This proves useful when we need to save time but is not necessary in cases in cases like ours where the data set is relatively small.

**Analyzing Model Performance**

**Specification**

Student correctly identifies the relationship between the training and test error as training size increases.

Meets Specification

**Specification**

Student provides analysis for both max depth 1 and 10 learning curve graphs. Both graphs have ample explanation if they suffer either high bias/underfitting or high variance/overfitting and are reasonably justified.

Meets Specification

**Specification**

Student identifies how the training and error curves relate to the increasing model complexity.

Meets Specification

**Specification**

Student clearly picks an optimal model from the model complexity graph with reasonable justification.

Does Not Meet Specification

**Reviewer Comments**

It is not clear why, given your correct description, you would choose 8 as a proper max depth instead of 5 as increasing complexity is not bringing any advantage though it is increasing the chance of overfitting. Please note as well that the training error is actually decreasing from 5 to 8 therefore your sentence: "the increase in max depth from 5 to 8 results in little change in test error or training error" is not correct.

**Model Prediction**

**Specification**

Student’s model gives a valid housing price with detailed model parameters (max depth) reported.

Does Not Meet Specification

**Reviewer Comments**

Please note that the in the project rubric it is stated: “Student’s model gives a valid housing price with detailed model parameters (max depth) reported.” Please extend your reasoning to max depth as well. You can achieve this by adding this code to retrieve the best parameter found by grid search:

print "Best model parameter: " + str( reg.best\_params\_)

**Specification**

Student compares prediction price to earlier statistics and justifies why the price is reasonable.

Meets Specification

**Reviewer Comments**

**Pro tip:** To assess if your prediction is reasonable, besides from comparing it with the median, the mean and checking if it is included in one standard deviation range, you could use SKlearn to find the nearest neighbours of the feature vector and see how your result compares with them.

from sklearn.neighbors import NearestNeighbors

def find\_nearest\_neighbor\_indexes(x, X): # x is your vector and X is the data set.

neigh = NearestNeighbors( n\_neighbors = 10 )

neigh.fit( X)

distance, indexes = neigh.kneighbors( x )

return indexes

indexes = find\_nearest\_neighbor\_indexes(x, X)

sum\_prices = []

for i in indexes:

sum\_prices.append(city\_data.target[i])

neighbor\_avg = np.mean(sum\_prices)

print "Nearest Neighbors average: " +str(neighbor\_avg)

<http://scikit-learn.org/stable/modules/neighbors.html#finding-the-nearest-neighbors>

**Additional Reviewer Comments**

Dear student,

well done with your good submission. There are only a few issues to be addressed in order to meet requirements, please refer to my comments in the appropriate section for some hints. I’ve left some Pro Tips as well in case you might be interested in learning more about some specific topics.

Keep up your good work!