UDACITY

Logc

‹ Back to Machine Learning Engineer Nanodegree

Predicting Boston Housing Prices

REVIEW | CODE REVIEW | HISTORY

Meets Specifications

Dear student,
You did a great project here. Congratz! To help you in the next module, I'd like to suggest this webpage:
http://usblogs.pwc.com/emerging-technology/machine-learning-methods-infographic/
It's going to be really useful :)
Cheers!

Data Exploration

All requested statistics for the Boston Housing dataset are accurately calculated. Student correctly leverages NumPy functionality to obtain these results.

All good here! Numpy is a versatile library and you are probably going to use it in all projects.

Student correctly justifies how each feature correlates with an increase or decrease in the target variable.

Your analysis looks correct, but it's only a guess based on your previews experience. Why don't you try to plot a graph (e.g., MEDV x RM) to confirm it? It's only a suggestion :)

Developing a Model

Student correctly identifies whether the hypothetical model successfully captures the variation of the target variable based on the model's R^2 score.
The performance metric is correctly implemented in code.

Student provides a valid reason for why a dataset is split into training and testing subsets for a model. Training and testing split is correctly implemented in code.

Good job! You set the random state, which allows us to reproduce the experiment. Besides that, it's really important to have a different test set to evaluate your model. Your points are correct.

Analyzing Model Performance

Student correctly identifies the trend of both the training and testing curves from the graph as more training points are added. Discussion is made as to whether additional training points would benefit the model.

Your analysis is correct. Great job!

Student correctly identifies whether the model at a max depth of 1 and a max depth of 10 suffer from either high bias or high variance, with justification using the complexity curves graph.

You are right. Besides that, it's always good to keep it as simple as possible. In this case, it converged to a good result with 300 samples and max_depth=3.

Student picks a best-guess optimal model with reasonable justification using the model complexity graph.

Yep, it looks like the best option among those options as well :D

Evaluating Model Performance

Student correctly describes the grid search technique and how it can be applied to a learning algorithm.

Right. Besides that, there are different kinds of GridSearch, like the RandomGridSearch (http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html#sklearn.model_selection.RandomizedSearchCV). It's useful when you want to try a lot of combinations (but it would take too long to try all). There are some papers showing that it is much faster and has similar results to GridSearch :)

Student correctly describes the k-fold cross-validation technique and discusses the benefits of its application when used with grid search when optimizing a model.

The cross-validation is an important technique for Machine Learning. It's really important and useful, mainly when we don't have a lot of data (and really useful to avoid overfitting).

Student correctly implements the `fit_model` function in code.

Student reports the optimal model and compares this model to the one they chose earlier.

Student reports the predicted selling price for the three clients listed in the provided table. Discussion is made for each of the three predictions as to whether these prices are reasonable given the data and the earlier calculated descriptive statistics.

Student thoroughly discusses whether the model should or should not be used in a real-world setting.

Yep, it's not useful nowadays, the model is no robust and there are many other things that may impact in the price. You are all right :)

DOWNLOAD PROJECT

RETURN TO PATH