UDACITY

Back to Machine Learning Engineer Nanodegree

Predicting Boston Housing Prices

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Dear student,
great project! Congratulations :D You've shown good knowledge with machine learning techniques.
I'd like to share this website with you (http://usblogs.pwc.com/emerging-technology/machine-learning-methods-infographic/). It's really interesting and might be useful for the next module.
Cheers!

Data Exploration

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

Yeah, it's working fine but it was required to use the Numpy lib to find the values. I'm sure you can fix it pretty fast, but you can check the Numpy documentation if you have a problem.
Here is the `amax` function: https://docs.scipy.org/doc/numpy-1.10.4/reference/generated/numpy.amax.html

I'm not going to ask you for a new submission because this one was the only issue and it's not a major one :)

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

Your intuition looks correct, but I couldn't see the plot :(

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.

Great job here! The R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression.

We can say that the R-Squared is the percentage of the response variable variation that is explained by a linear model and it's always between 0 and 100%:

- 0% indicates that the model explains none of the variability of the response data around its mean.
- 100% indicates that the model explains all the variability of the response data around its mean.

So, the higher the better. In this case, we can explain 92% of the variability, so the model is supposed to fit well.

Besides the number of samples, there are more things that we should consider when using the R2 score (e.g., it should be a linear regression to make sense), but it's not going to be discussed here :P

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. And your explanation about the training and testing set is pretty straightforward.

Using the test set we can check the performance of the model in unseeing data, so we can try to check if it's really generalizing the problem. If there's no test set, we can't check how it's performing with independent data, so we would be blind. The training score is important to check if the model is learning something, but it doesn't really mean that the model would really perform well with unseeing data.

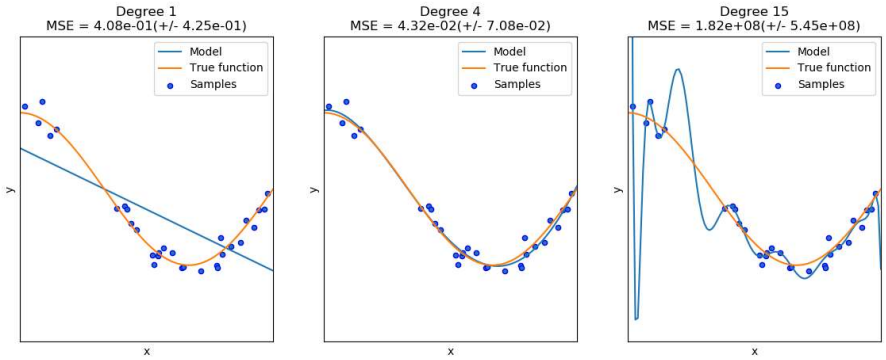
Analyzing Model Performance

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 observation is correct. In this case, adding more training samples wouldn't improve the model.

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.

Perfect! Those concepts are really important, but without the training/testing sets, it wouldn't be possible to identify.
I really like the Sklearn documentation (and I check it almost every day). It has some awesome examples. I'd like to share this one specifically with you:

> Underfitting vs. Overfitting
>
> 
>

Besides that, the `max_depth=10` is overfitting case. We can think "Oh, it has an awesome training score! It's really learning something!", but if we don't check the test score, or model would be faded to fail.

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

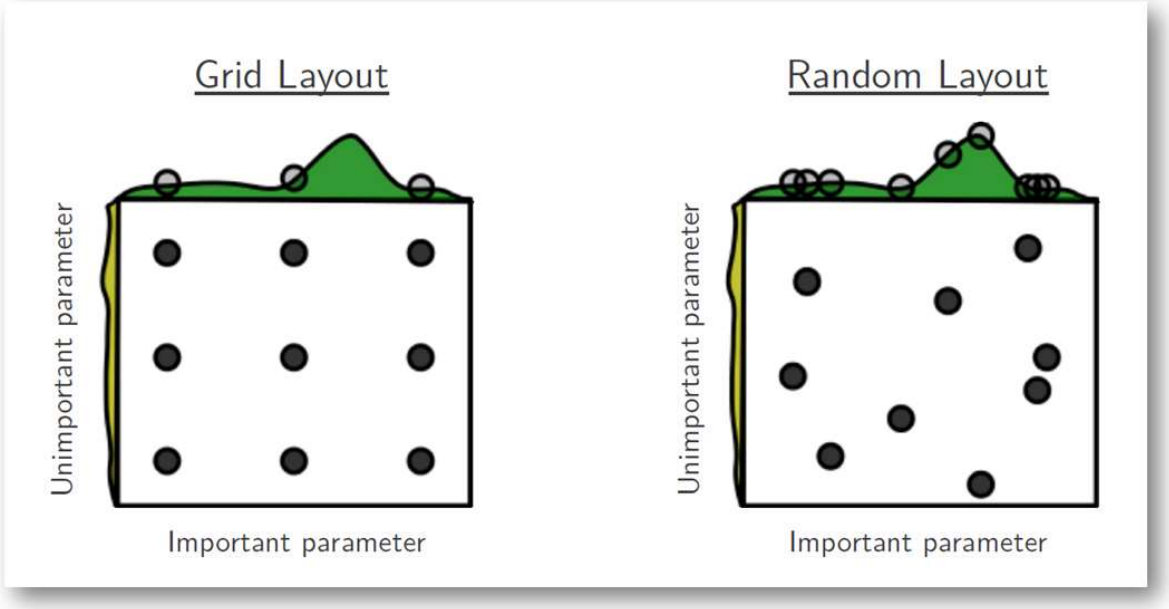
With `max_depth=1` and `max_depth=3` the model is converging, but only the `max_depth=3` has a better score. With the higher depths, it's not even converging (and it's really overfitting). Maybe, with much more samples, it would converge for the `max_depth=6` and `max_depth=10`, but we can't state that without trying. `max_depth=4` seems okay to me.

Evaluating Model Performance

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

Right. But remember: the Gridsearch perform an exhaustive search, what can be really expensive (computationally speaking). 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 :)

Example:



Source: https://medium.com/rants-on-machine-learning/smarter-parameter-sweeps-or-why-grid-search-is-plain-stupid-c17d97a0e881

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.

You really understand the cross-validation idea. It's really important to perform this kind of technique to try to avoid overfitting.

Student correctly implements the `fit_model` function in code.

I'm glad that you have remembered to set a `random_state` number :D
But one tip here for a more pythonic code. You could use `range(1, 11)`. Here isn't a big problem, but think in more hyperparameter to tune (or larger ranges)... could be a HUGE issue :)

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.

Yes! You really shouldn't use this model nowadays. It's very old and a lot of things changed. And I agree, it's not robust enough. Great analysis here.

DOWNLOAD PROJECT

RETURN TO PATH