UDACITY

PROJECT

Predicting Boston Housing Prices

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW

CODE REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Requires Changes

2 SPECIFICATIONS REQUIRE CHANGES

Congratulations for a very strong submission! A couple of your answers require changes, but nothing major, and I'm sure your next submission will meet all specifications. Keep up the good work!

Data Exploration

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

Awesome

All statistics are calculated using `numpy`. Well done! :)

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

Awesome

Excellent job plotting the variables to inform your answer to **Question 1**!

Suggestion

Another way to visualize your data is using the seaborn package, which will throw in a regression line for good measure:

```
import matplotlib.pyplot as plt
import seaborn as sns

for var in ['RM', 'LSTAT', 'PTRATIO']:
    sns.regplot(data[var], prices)
    plt.show()
```

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.

Comment

Note that how good an R^2 score is really depends on the domain of the problem. In a well controlled environment such as laboratory testing, you would expect R^2 to be higher than in a "messy" study full of noisy data, uncertainty regarding measurements, etc.

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.

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.

Comment

> This is a overfitting scenario when the training algorithm don't go well with the testing set, but it goes well with the training data. So, increasing more training points would not benefit the model.

Note that, in high variance (or overfitting) situations, it is possible that more data will help the model out. It's usually harder for a model to perfectly fit a larger training set, which will tend to present more variation. We can see this in the learning curves for a maximum depth of 1 and 3, which are overfitting at first but whose variance pretty much disappears after more training points are added.

In this case, however, it seems like the testing score has plateaued around the 0.6-0.7 mark (for a maximum depth of 10), indicating more training points will not help out.

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.

Comment

> With maximum depth equals 10, the model is starting to suffer with high variance. The training score is almost perfect, but the validation score isn't above 0.8.

I would say the model starts suffering from high variance well before a maximum depht of 10. When the maximum depth goes from 4 to 5, for instance, we see a significant increase in the training score and a slight decrease on the testing score, indicating the start of an overfitting problem.

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

Awesome

Good call!

Evaluating Model Performance

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

Required

Two important bits are missing in the answer:

- Given the possible values for each hyperparameter, how many hyperparameter combinations are analyzed by grid search?
- How does grid search "try out" different hyperparameter combinations? What does it *do* with these combinations to find out which is the best one?

Hint: why can randomized search be preferred to grid search in some situations?

Comment

I use **hyperparameter** here to refer to the parameters that the model does not learn from the data, as these are the parameters that we will define with the help of grid search.

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.

Required

- The main benefit of using cross-validation along with grid search is missing from the answer.
- Think about the alternative of using a single validation set to fit all instances of the model in a grid search, considering all possible parameter combinations.
- When we pick the best instance of the model at the end of this validation process with a single validation set, what do we have?
    - A model that generalizes well to unseen data?
    - Or a model that *fits the validation set well*?
- And how can cross-validation help us here?

Comment

Just a quick note to make sure you understanding we are dealing with three different subsets of our data:

- The **training set** is used to train the model - i.e., to find the model's *parameters* (the ones learned from the data).
- The **validation set** is used to calibrate the model - i.e., to find the model's *hyperparameters* (the ones not learned from the data).
- The *testing set* is used to evaluate the model - i.e., to check how the model's final instance (trained and calibrated) performed when given previously unseen data.

Cross-validation techniques allow us for using the same set for both training and validation, but the testing set must remain held out until a final instance of the model is defined. You can find more on how the train-validation-test process works here.

Student correctly implements the `fit_model` function in code.

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

Comment

I believe that, if the score for models with maximum depth of 3 and 4 is close (as suggested by the model complexity curves from questions 5 and 6), there is an argument for considering the difference is due to noise and keeping the simpler model.

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.

Awesome

Great job using the previously calculated statistics to analyze whether the predicted prices make sense for these sample clients.

Suggestion

To go above and beyond on this question, you could also compare the features of the three examples to some statistics of the distribution of features for the whole dataset. You can check these statistics by running `features.describe()`, which will print out the five-number summary, the mean, and the standard deviation of each feature.

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

Comment

> The features to construct the model should be relevant. If there are a lot of feature, the model will be complex and to train the model will request a lot of data. And if there is few of them, the algorithm would not be capable to capture the patterns on the data.

- Adding **meaningful features** to our model may be a good way of reducing the so-called **irreducible error**, or the error that comes from variation in our target that our predictors are not able to explain.
- More predictors (as long as they have a true relationship with the target that is not captured by the other predictors in our data set) will give useful information to our model, which can use this information to improve its predictions.
- You can read more about the irreducible error, bias, and variance in the resources linked in this post from the project's discussion forum.

RESUBMIT

DOWNLOAD PROJECT

Learn the best practices for revising and resubmitting your project.

Have a question about your review? Email us at review-support@udacity.com and include the link to this review.

RETURN TO PATH

Student FAQ