UDACITY

Logout

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW | CODE REVIEW | NOTES

Meets Specifications

SHARE YOUR ACCOMPLISHMENT

Congratulations to build a high-quality project! You have done a great job. Keep it up, and enjoy learning!

Data Exploration

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

Well done on fixing the code and getting the statistics using `numpy`. For most real world projects, getting statistics of the data is the first step to understand the data before subsequent feature engineering can take place. You may also examine if there are any missing values and outliers.

There are other statistical measures supported by `numpy`. For example, you may also look at the percentile of the data:

```
first_quartile = np.percentile(prices, 25)
third_quartile = np.percentile(prices, 75)
```

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

Your reasoning makes perfect sense. You can also visualize the features vs. prices:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(20, 5))
for i, col in enumerate(features.columns):
    plt.subplot(1, 3, i)
    plt.plot(data[col], prices, 'o')
    plt.title(col)
    plt.xlabel(col)
    plt.ylabel('prices')
```

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.

Great job!

Setting the `random_state` is very useful, and should be done in almost all cases, as it ensures the results are consistent across multiple runs in cases where consistency is desired. For example, to compare performance of multiple models / configurations, we often want to evaluate the performance based on the same training and test sets.

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.

"after 100 points, the model benefits only very minimally from more data":
Absolutely. As we have more data, we can generalize better on the test set, up to certain limit, where more data does not really help much.

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.

There is usually a trade-off between bias and variance. Here is a more detailed reference that I find very useful:
http://scott.fortmann-roe.com/docs/BiasVariance.html

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

As an interesting note, you may also read about Occam's Razor principle, which favours a simpler model to more complicated ones.

Evaluating Model Performance

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

"Grid search exhaustively tests all parameters in a specified subset of the parameter space, thus finds what are the optimal parameters that yield the best results":
Awesome description 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.

Well done to explain k fold CV.

K-fold CV does have its limitations:

- it is more computationally expensive than hold out method
- it does not work well when data is not uniformly distributed (e.g. sorted data).

Student correctly implements the `fit_model` function in code.

Great job to implement model tuning!

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

For thought: what could cause the result of fit_model by grid search and the initial guess in Q6 to be different?
Hint: How the data are split in these two cases?

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

Fantastic analysis of the result. This is really important, as in real-world projects, often we are not only required to get the result, but also to explain the result as well.

You can also visualize the result by overlaying the prediction with histogram of the data as:

```
import matplotlib.pyplot as plt
plt.hist(prices, bins = 20)
for price in reg.predict(client_data):
    plt.axvline(price, lw = 5, c = 'r')
```

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

DOWNLOAD 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