UDACITY

Logout

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW | CODE REVIEW | NOTES

Requires Changes

SHARE YOUR ACCOMPLISHMENT

5 SPECIFICATIONS REQUIRE CHANGES

Very nice analysis. One tip here would be that some of these topics are extremely important as you embark on your journey throughout your Machine Learning career and it will be well worth your time to get a great grasp on these topics before you dive deeper in. Keep up the hard work!!

Data Exploration

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

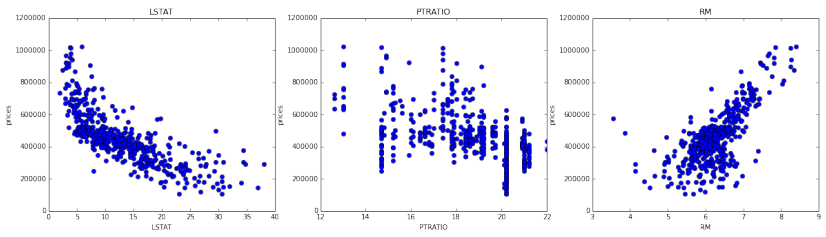
You have the correct ideas here. However for this section please calculate these statistics with numpy. As you are actually using pandas `Series` here. Therefore I bet numpy has the same commands

```
minimum_price = np.min(prices)
....
```

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

Nice observations for the features in this dataset. As we can confirm these ideas by plotting each feature vs MEDV housing 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.

Nice discussion here. We can clearly see with this high r^2 score of 92.3% (0.923) that we have strong correlation between the true values and predictions. Would also recommend stating the optimal score would be 1 or 100% for a more solid answer.

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.

Nice with your comment of "*This allows you to verify whether your model generalizes to new data, and doesn't just work with data that it has seen before.*" As we can get a good estimate of our generalization accuracy on this testing dataset. Since our main goal is to accurately predict on new unseen data. Also note that we can protect against overfitting with this independent dataset.

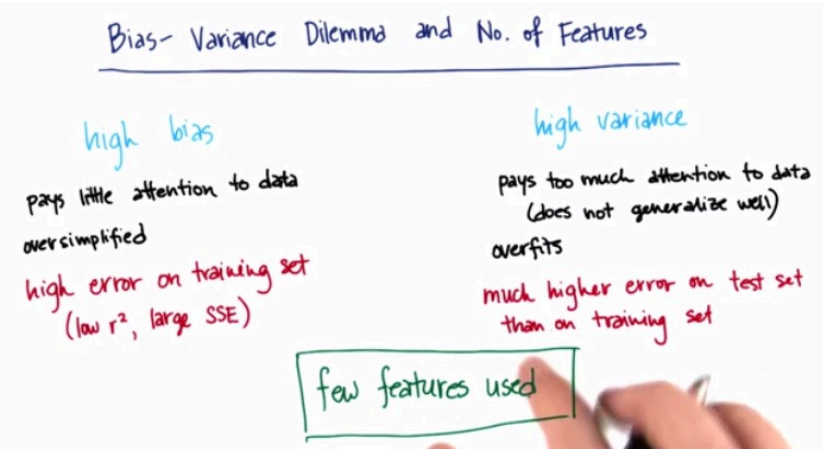
Analyzing Model Performance

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.

Nice description of the training and testing curves here. However relook at the comment of " *In this scenario, more training points benefits the model by making it more precise, and preventing underfitting.*" This really isn't correct by looking at the testing curve. Therefore it is actually clear that the testing curve has converged to its optimal score(no more benefit with increased data) so more data would not benefit 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.

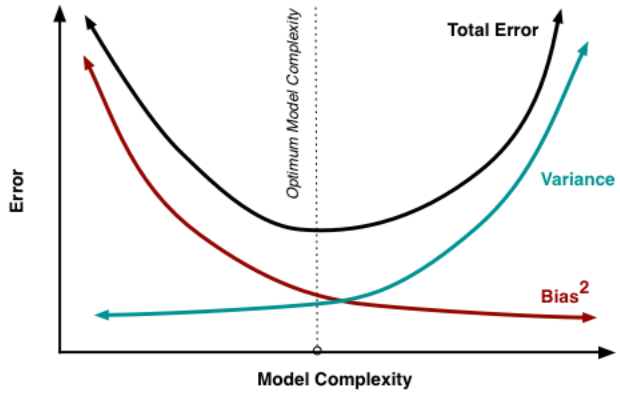
You are correct that a max_depth of 1 suffers from high bias and a max_depth of 10 suffers from high variance. However not too sure with your visual justification. As you comments of "*since the score of around .7 is less than the scores at lower complexities*" and "*the testing score is very low at around 0.3 compared to the highest value of around 0.78 at depth 5*" wouldn't really represent high bias or high variance. As you should be discussing both the training and validation score for both of these. Therefore for a max depth of 1 and a max depth of 10, what are the training and validation scores? And how do they compare to each other? Are they high? low? close? far apart? etc...



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

" *it has the highest R^2 score for its testing sample*"

Good justification for your choice. We are looking for the highest validation score while considering the best bias / variance tradeoff.



Evaluating Model Performance

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

You are correct that GridSearch try to " *find what are the optimal parameters that yield the best results*." Lastly can you also briefly mention which hyper-parameter value combinations does it test (e.g. a random sample of them, every other combination, all of the exhaustively)? What are these "*different parameters*"?

Links

- (http://scikit-learn.org/stable/modules/grid_search.html)
- (https://en.wikipedia.org/wiki/Hyperparameter_optimization#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.

Need a couple minor things here

- Make sure you also mention the last step in the k-fold cross-validation technique. What do we do with all the error rates at the end?
- Can you also expand on your discussion for the benefits of its application when used with grid search. Therefore can you elaborate on your comment of "*all of the data can be used for both training and testing, instead of allocating a fixed amount as test data.*" Why is this true? What would essentially happen if we were to do parameter tuning on only one data split?

Links

- (https://en.wikipedia.org/wiki/Cross-validation_(statistics)#k-fold_cross-validation)
- Video
- (https://www.cs.cmu.edu/~schneide/tut5/node42.html)

Student correctly implements the `fit_model` function in code.

You have correctly implemented GridSearch!! Just note that we can also pass in the param grid with the `range` command

```
params = {'max_depth': range(1, 11)}
```

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

Nice work!!

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.

Good. Comparing these predicted prices to the mean/std/min/max of the housing prices and the features are a fair assumption on why the prices seem reasonable. Just remember to keep in mind the error rate for the test set here.

Pro Tip: We can also plot a histogram of all of the housing prices in this dataset and see where each of these predictions fall

```
import matplotlib.pyplot as plt
plt.hist(prices, bins = 30)
for price in reg.predict(client_data):
    plt.axvline(price, c = 'r', lw = 3)
```

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

I almost seems like you wouldn't use this model based on your analysis here. This dataset is quite old and probably doesn't capture enough about housing features to be considered robust!!

RESUBMIT PROJECT

DOWNLOAD PROJECT

Best practices for your project resubmission

Ben shares 5 helpful tips to get you through revising and resubmitting your project.

Watch Video (3:01)

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