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

Please note that the final HTML submitted is not aligned with your iPython Notebook, please kindly have a check :)

Data Exploration

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

Please look at the code as following which would be getting the corresponding values in the client's feature set in a programmable manner:

```
chosen_features = ['LSAT', 'RAD', 'TAX']
features = city_data.feature_names.tolist()
for feature in chosen_features:
    index = features.index(feature)
    print CLIENT_FEATURES[0][index]
```

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

Please note that this project has been switched to a new project specification. As this is a resubmission - our review is still stoiked with the old specification

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.

This question is required for new specification

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.

SEEDING YOUR ALGORITHMS:

- In order to remove randomness of your algorithms, and to make sure your results don't differ at each run, please consider to always use a random seed to seed your algorithms.
- A standard practice I've come across is to define a random seed as a global variable in your work, and to use it throughout all the algorithms/methods which require random number generation (splitting data, decision tree initialisation, neural network weight initialisation etc).
- In sklearn, as far as I know, random seeds are provided to methods and functions using the parameter `random_state`. Please seed all of your algorithms in the future if you haven't been doing so yet

Analyzing Model Performance

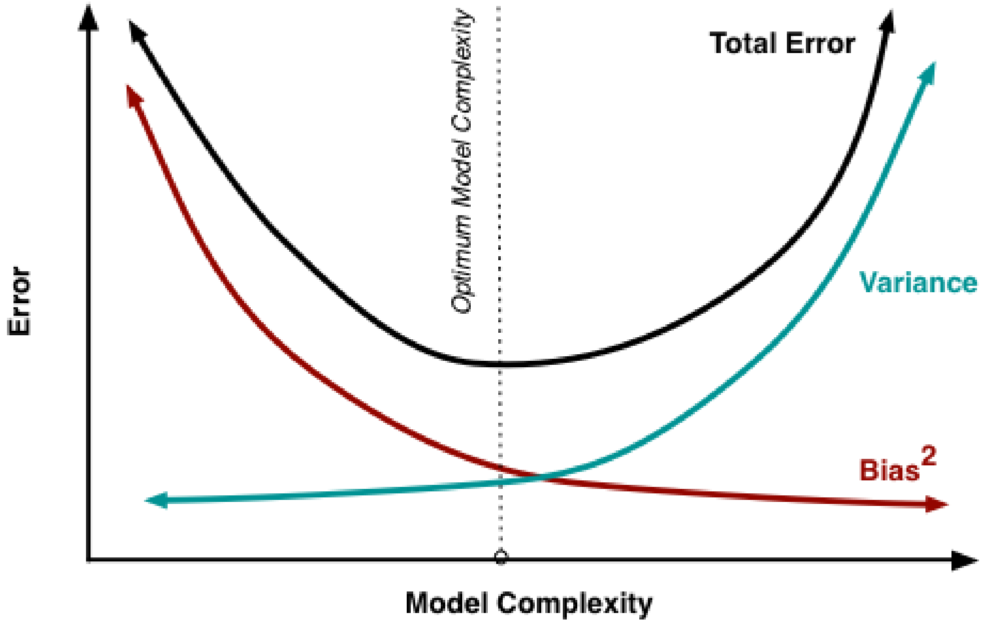
Student correctly identifies the trend of both the training and testing curves from the graph as more training points are added. Discussion is made as to whether additional training points would benefit the model.

- There are two phases to the error rates - the testing error approaches some form of an asymptote and begins perturbing while the training error continues to increase.
- To give more context, when the training set is small, the trained model can essentially “memorize” all of the training data. As the training set gets larger, the model won’t be able to fit all of the training data exactly.
- The opposite is happening with the test set. When the training set is small, then it’s more likely the model hasn’t seen similar data before. As the training set gets larger, it becomes more likely that the model has seen similar data before.

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.

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

Please look at the following diagram, which clearly shows how the testing and training error evolves with the increasing model complexity



- Please note that with the increasing model complexity, the model goes through two stages from underfitting to overfitting - Please consider to include this in your report

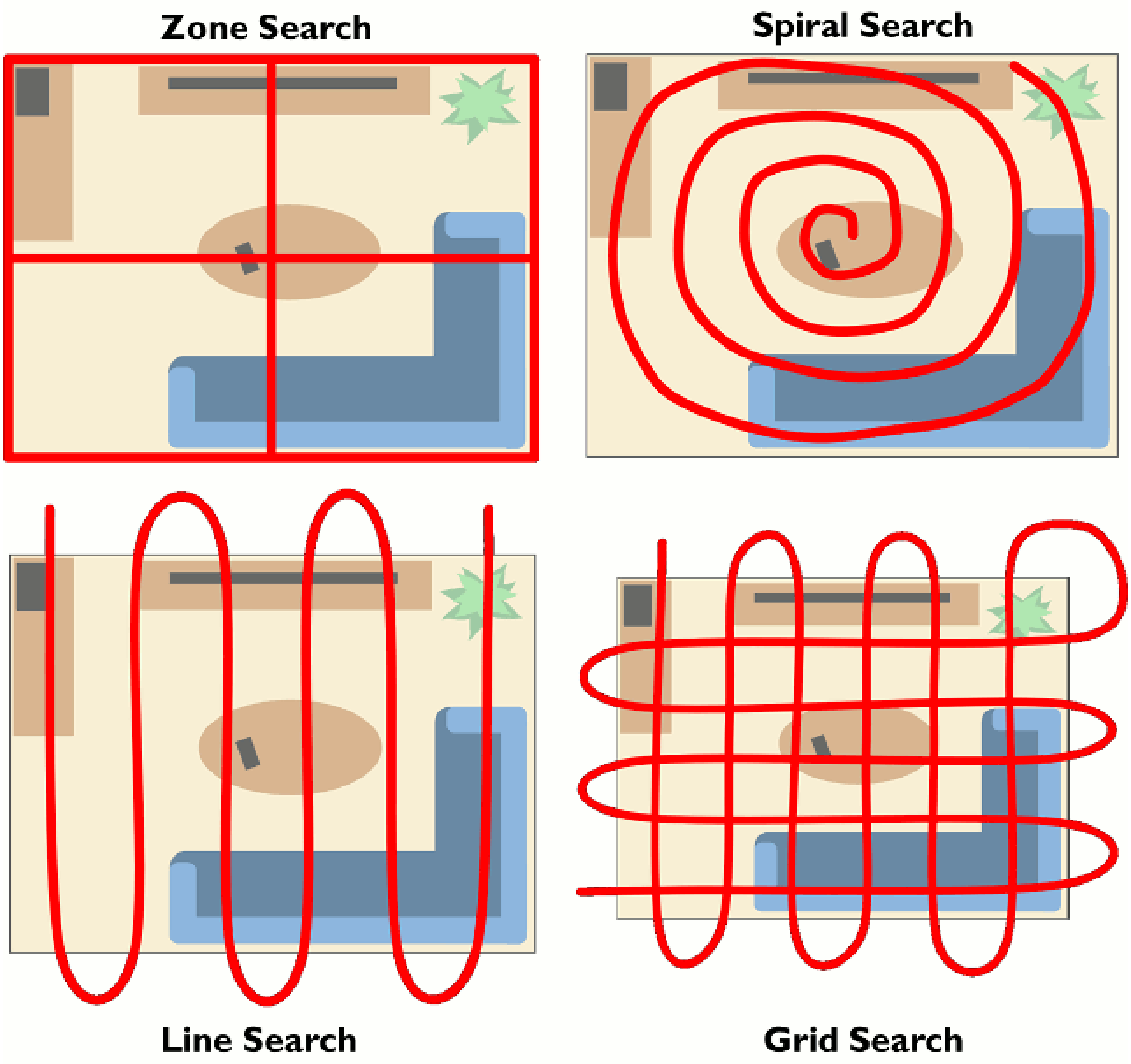
- The first phase is where the model is underfitted and the training error is exceedingly high.
- The second phase is where the model is overfitted and the difference between testing and training error is high.
- The optimal model is where the turning point at, which the training error is low and testing error is at global minimum.

Max_depth = 5 would be the best choice since the training and testing error are both low and the model complexity is moderate (higher value of max_depth would lead to the overfitting)

Evaluating Model Performance

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

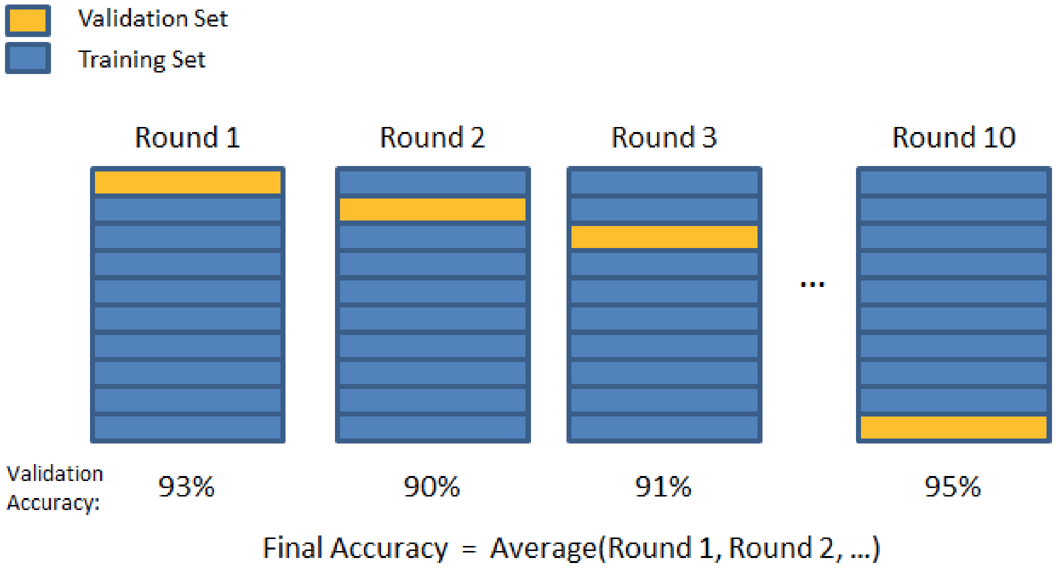
- Grid search algorithm which is simply an exhaustive searching through a manually specified subset of the hyperparameter space of a learning algorithm. A grid search algorithm must be guided by some performance metric, typically measured by cross-validation on the training set
- It would be worth mentioning about fine tuning a learning algorithm for a more successful learning/testing performance in terms of the application for grid search.
  Please look at the following comparison for different space 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.

- Please note that the cross validation specification is over simplified.
-
    - Please consider to elaborate more regarding to how cross validation helps with the grid search.
- Kindly note that for Cross validation, especially on K-fold CV - the data set is divided into k subsets, and the holdout method is repeated k times. Each time, one of the k subsets is used as the test set and the other k-1 subsets are put together to form a training set. Then the average error across all k trials is computed. The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once, and gets to be in a training set k-1 times.

  Please look at the following diagram which clearly shows how the k-fold CV works:



- Based on the above description, you may notice that Cross validation is useful because it maximize both the training and testing data so that the data we can use to provide best learning result and best validation - this is extremely useful when the dataset is limited in size - please think about how does help with grid search? (making full use of the dataset)
- If we limit the grid search in a single dataset, would overfitting possibly happen?

Student correctly implements the `fit_model` function in code.

As in this project we would like to focus on only the max_depth's optimal value, thus please consider to only focus on the max-depth for grid search, i.e.

```
parameters = {'max_depth':(1,2,3,4,5,6,7,8,9,10)}
```

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 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.

It would be better that could consider the following questions:

- Would additional data points (or the inclusion of data per year) benefit the model? Please look at the dataset for when the data is collected.
- Is there a possibility of outliers in the data that can drastically change predictive results?
- Does this dataset feature enough characteristics about homes to be considered robust?
- Does performing grid search on the entire dataset affect your confidence in the model? There are two important factors that you could make use for consideration:
- Data Quality: Please access the data quality, whether the data is recently collected or outdated; Do you think the features is enough for the data points collected is enough?
- Model Training: Please review the algorithm we use, do you think there are better ones? Do you think grid search on the entire dataset affect the accuracy?

☑ RESUBMIT

DOWNLOAD PROJECT

Student FAQ

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

Rate this review