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

1 SPECIFICATION REQUIRES CHANGES

Quality of Code

Student's code runs successfully and produces results similar to those in the report. No modifications are made to the template code beyond what is requested without justification.

Statistical Analysis & Data Exploration

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

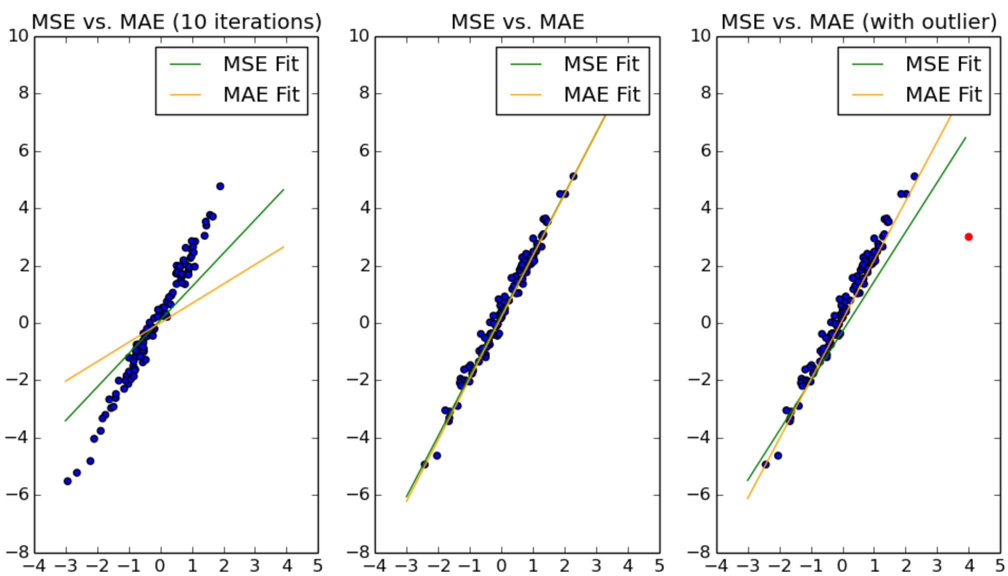
Great job here! Looking at some basic descriptive statistics before we dive deeper into our data can help us steer clear of assumptions that might not necessarily apply (e.g. assuming the data is normally distributed when in fact it is not). It also help us get a feeling for what a sensible prediction value might be.

Student adequately describes three separate features of the dataset. The corresponding values in the client's feature set are correctly identified for the chosen features.

Evaluating Model Performance

An appropriate performance metric is chosen with thorough justification. The metric is correctly implemented in code.

Both MSE and MAE are appropriate. There are instances however where one error metric can help us achieve better performance. Below is an example where a linear model has been trained on some synthetic data by means of stochastic gradient descent based on both MSE and MAE:



Note that MSE converges faster to the solution while MAE is more robust to outliers. Notice that although MSE punishes larger errors more heavily than MAE they both guide the model to almost the same solution if no outliers are present and enough iterations are performed.

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.

"To evaluate performance of model generated from train data"

Please explain in more detail why is this model evaluation improved by having two separate training and testing data sets.

Student correctly describes the grid search algorithm and briefly discusses its application. GridSearchCV is properly implemented in code.

"Uses grid of parameters to fit a model and picks the best one. It is used in model parameter tuning with exhaustive search."

Correct! Grid search is essentially an exhaustive search as you describe, well done!

Student correctly describes how cross-validation is performed on a model, and why it is helpful when using grid search.
Modifications beyond the default 3-fold cross-validation for GridSearchCV are reasonably justified.

"Cross-validation is a model validation technique avoids the possible bias introduced by relying on any one particular division ((conventional validation) into test and train components , instead partition the original set in several different ways(multi-fold) for good distribution, spread of samples and computes an average score over the different partitions"

Well explained!

Analyzing Model Performance

Student correctly identifies significant qualities of the training and testing errors as the training set size increases.

"Max depth 1. Training eror incrased intially and stabilized after 100 samples of trainingging data. Testing error too reduced dramatically in the begging and stabilized around 100 samples."

Good observations! Training error increases as the model is no longer able to fit the data perfectly, training error decreases and eventually plateaus.

Student provides analysis for both a max depth of 1 and a max depth of 10. Reasonable justification is given for each graph if the model suffers from high bias or high variance.

"When max depth is 1 it suffers from high bias. High error value and quick convergence in training and

test set indicate underfitting. For max depth 10, I think it sufferes over fit (high variance) due to no change (very low/no error) in training error with increased data set and there still high testing error."

Clear and accurate. Arguments are supported by observations made on the plot, great job!

Student identifies how the training and error curves relate to increasing the model's complexity.

"As max depth increases training errors are reducing however, testing errors reduce until maximum depth 4(~5)."

Accurate, well done!

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

Model Prediction

Student determines the optimal model from parameter tuning and compares this model to the one they chose.

Student's model produces a valid result. The predicted selling price is adequately justified by the calculated descriptive statistics.

Student thoroughly discusses justification for or against using their model for predicting future selling prices.

RESUBMIT

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

Rate this review