UDACITY

PROJECT

Predicting Boston Housing Prices

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Meets Specifications

The project has met all the specifications. Hope you enjoyed working on this project. All the best! 👍😃

Data Exploration

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

The requested statistics have been calculated correctly and good use of NumPy functionality has been made!

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

Apt justification has been provided as to how a feature correlates with an increase/ decrease in the target variable.
Suggestion - Another way to verify our reasoning would be to plot each feature against MEDV housing prices and fit a regression line. You can use this blog post as a reference.

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.

Good explanation has been provided in regards to how a model's R-squared score captures the variation of the target variable.
Suggestion- Apart from R-squared score, residual plots are a good way of checking out regression problems. You can get more information on this topic by visiting this link

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.

Valid reasoning has been provided as to why a dataset is split into train and test sets.
Suggestion - To learn more about why we require to split our data into train and test sets please check out video1 and video2 from Udacity. Also visit this blog post for more information on the topic.

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.

Great job on the analysis here.
Suggestion - To add to the discussion, if we observe the training score curve: when the number of training points are close to 0, the training score is close to 1. This is because the model overfits and we see that as more training data points are added, training score drops to around 0.9 because the model cannot explain all the variance anymore. In terms of the testing score curve, when the number of training points are close to 0, the testing score is close to 0. This is because the model overfits and doesnt generalize well and we see that as more training points are added, testing score increases rapidly and converges to a value also close to 0.7.

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.

Well done! the reasoning for the complexity curves is absolutely correct.
Suggestion - You can refer to the following links 1,2 and 3

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

Good intuition has been provided in selecting the best guess optimal model 👍

Evaluating Model Performance

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

Great job on clearly understanding this technique.
Suggestion - Note that as gridsearch is an exhaustive search it is computationally very expensive while dealing with a large number of hyperparameters and larger datasets. You clearly understand this technique. As you can see that we are using a decision tree and max depth in this project. Can also note that since this is an "exhaustive search", one limitation of GridSearch is that it can be very computationally expensive when dealing with a large number of different hyper-parameters and much bigger datasets. Therefore,it is worthwhile exploring RandomizedSearchCV technique in order to validate our hyperparameters.

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.

Great job on correctly discussing KFold cross validation

Student correctly implements the `fit_model` function in code.

The fit_model has been correctly implemented 👍

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

Here note that GridSearch in combination with CV searches for the model with highest validation score on the different data splits made by using ShuffleSplit.

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.

Good job on comparison of the three predictions. To analyze this in more depth, histograms of all the housing prices can be made in order to see where these predictions lie or the descriptive stats of the features can be used for comparing the three features using describe()method.

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

I agree with the reasoning provided. As the dataset is relatively old, it doesn't take into account the modern housing features, and the range in predictions is also quite large, thus the model cannot not be considered robust.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

Student FAQ