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

11 SPECIFICATIONS REQUIRE CHANGES

Your answers show the great amount of effort you have put into learning the concepts. Just a few concepts to pick up/additions to be made in your answers and you will be good to go ahead on your journey to becoming a Machine Learning Expert. All the best. Happy learning. 👍 😄

Data Exploration

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

Great work getting the summary statistics. However, the purpose of this section is to introduce you to the efficient numpy package. Please use the functions available in numpy package and rewrite the respective code lines.

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

Correct. You have captured the concept of correlation accurately here. There are many types of correlation in statistics actually. Do you know what kind of correlation you have described in your answer? Check out the following link to find out:
http://www.statisticssolutions.com/correlation-pearson-kendall-spearman/

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.

Why do you think so? Try focusing on defining what R^2 means, how is it calculated, what is the range of values it can assume and then compare the output with the range to justify your answer.

R-squared value is the percentage of the response variable variation that is explained by a linear model:

R-squared = Explained variation / Total variation

In the target variable, there is some variation contained. The model is trained to learn pattern from the independent variables and try to explain the variation in target variable.

Total sum of squares (SST)= sum of squares explained by regression model(SSR) + sum of squared errors(SSE)
1 = (SSR/SST) + (SSE/SST)
(SSR/SST)=1-(SSE/SST)
R^2 = (SSR/SST) =1-(SSE/SST)

Please modify your answer to demonstrate this understanding.

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 implementation. How ever, the train/test split has no definitive impact on whether a model will over fit or underfit.
Consider the following example:

How good will be the learning for a model totally depends on how much relevant information is contained in the data set. DO you think a train data set with 80,000 rows and test data of 20,000 rows result in a better trained model? How about 70,000 and 30,000? Well, we can't say anything about that definitively.

What if the data is about some sensor readings and it senses environment 100 times in a second. So for one second of time, 100 rows would be present. That means 100,000 (80,000+20,000) would represent 1000 seconds data only which means data of about 16.66 minutes. Would you expect this model trained on ~16.66 minutes to perform well 24x7x52 (round the year?). Obviously not. Here we need more data to get a better trained model and not a good split or a split at all!

When it comes to measuring generalization, we do it by observing the model performance on a data set which has not been seen by the ML model. When we are talking about the model evaluation on the unseen data, we would need the true values for that unseen data for us to compare the model predictions with. If we train the model on the whole available data set, would we be able to check its performance on unseen data set? Modify your answer to reflect above mentioned points.

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.

Your answer is giving me a feeling that you think that the model error is plotted on Y-axis. But if you see closely, Y-axis represents the model score (performance) and not the error. In the light of this fact your answer should change. Also, after a particular value of number of training points, the training and testing curves have flattened out. That means for any number of training points, the score would remain the same. Do you think adding more points would help given the above observation?

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.

Same observational error has occurred here. Y-axis represents the model performance and not the error. This instance reminds me of a popular quote:
"Analysis is objective but the interpretation is subjective!"

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

Due to error in recognising that Y-axis represents the score, this answer also needs a relook from you.

Evaluating Model Performance

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

Your answer suggests that you are associating F1 score with grid search very closely.

Please understand that F1 score is the harmonic mean of precision and recall. Precision and recall are found out from confusion matrix which itself is found out for classification problems only. Thus, F1 score as a model evaluation metric becomes applicable only for classification problems.In general, depending on the type of

problem, regression or classification, the evaluation metric is chosen. Please modify your answer to reflect this understanding.

Also, I am sure you do understand the difference between parameters and hyper-parameters as these two concepts are different and the terms are not interchangeable.

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 explanation. There are certain finer details of this concept, which you should know and which should be part of your answer. Let me explain as these are the points generally asked in Data Science / Machine Learning interviews.

K-fold CV is an algorithm validation technique: whether a given algorithm will train properly or not. When you get different models from different folds, what you do is average out the evaluation metric of all the models to get what?
Well, to get an 'unbiased estimate of model generaliztion on unseen data'. That is the main purpose of k-fold cross validation. Once we are satisfied that a particular algorithm is good for a particular data set, we test it on the dataset we had kept aside at the beginning.

A bit about k-fold using an example:
Let us assume we have 100 rows in total. Out of these 100 rows, let us say 20 rows are taken out for final testing of the model and 80 rows are kept for training. Now these 80 rows are divided into k-folds. If we assume k to be 10, then each fold consists of 8 rows. Now out of these folds, one fold having 8 rows is kept as validation and rest 72 rows are used as training. This process is repeated with each fold being one validation set. Once we are sure about our model, the final model performance is check on the 20 rows we had taken out in the beginning.

Its application on grid -search:
We take one hyperparameter combination from the grid and keep it constant for one round of whole k-fold cross validation process: the whole splitting into kfolds, training and validating on one fold and so on. This helps us in getting an unbiased estimate of model evaluation metric which helps us decide in a more unbiased way, whether the given combination of hyperparameters is best for the particular data set or not. This process is repeated for all combinations of hyperparameters in the grid.

Using the train-test split method to evaluate model or choose best hyper parameters has a risk. The data set in the fixed train-test method may result in optimum hyper-parameter combination which is good for only that particularly arranged training dataset. To get rid of this bias towards a specific arrangement and composition of data, k-fold becomes useful.

Student correctly implements the `fit_model` function in code.

Great

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

Your expectation about max_depth = 4 not being a good value for the parameter would change when you read the evaluation curves and complexity curve considering Y-axis as the score and not the error.

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.

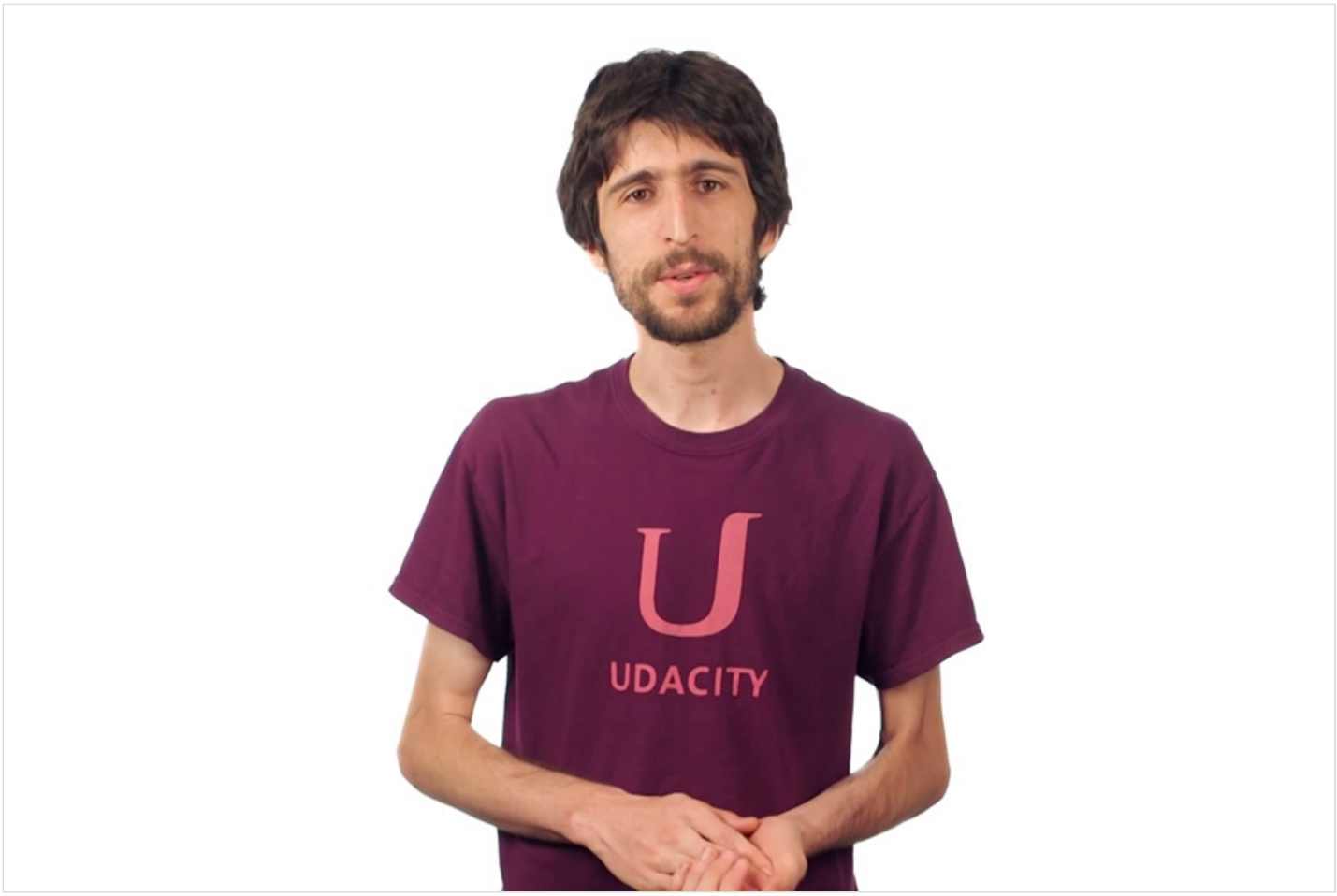
Great work considering the correlation with the feature values. This answer will become perfect when you also blend in the information about the summary statistics which you calculated in the very first code block.

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

Great answer. You would need to change your third points whne you read the graphs correctly.

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)

RETURN TO PATH

Rate this review

Student FAQ