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

3 SPECIFICATIONS REQUIRE CHANGES

You did a pretty impressive job in your first submission for this project. You showed good coding skill and solid understanding of the concepts we taught. However, project feedback is the most important learning part here at Udacity. I'm expecting your next submission.

Data Exploration

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

It's always the first step to explore the dataset before implementing any machine learning algorithm. Here you calculated the statistics using `pandas`, but we require students to calculate using NumPy e.g.: `np.std(prices)`.

Pandas' default standard deviation is different from that of numpy. Here we need to calculate the std of population, not samples. You can check the difference:

https://docs.scipy.org/doc/numpy/reference/generated/numpy.std.html
http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.std.html?highlight=std#pandas.DataFrame.std

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

Excellent reasoning and visualization. We can also check whether our analysis matches the machine learning algorithm result later in this project!

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.**

Well done! The R^2 score is one of the metrics to evaluate model performance. We will learn many more in the coming lesson. For R^2 you also may want to check Khan's video and some of its caveats.

https://en.wikipedia.org/wiki/Coefficient_of_determination#Caveats

Sklearn also has a dedicated page about all model evaluation methods.
http://scikit-learn.org/stable/modules/model_evaluation.html

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.

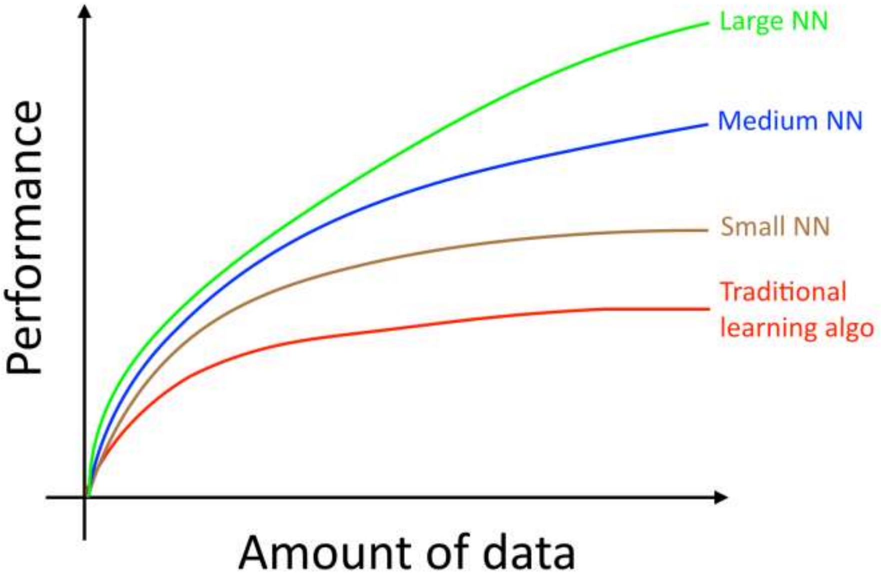
Excellent answer!

The ultimate goal of machine learning is to build predictive model that generalizes well on unseen data. Without testing set, we would never know how well/bad our model performs.

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.

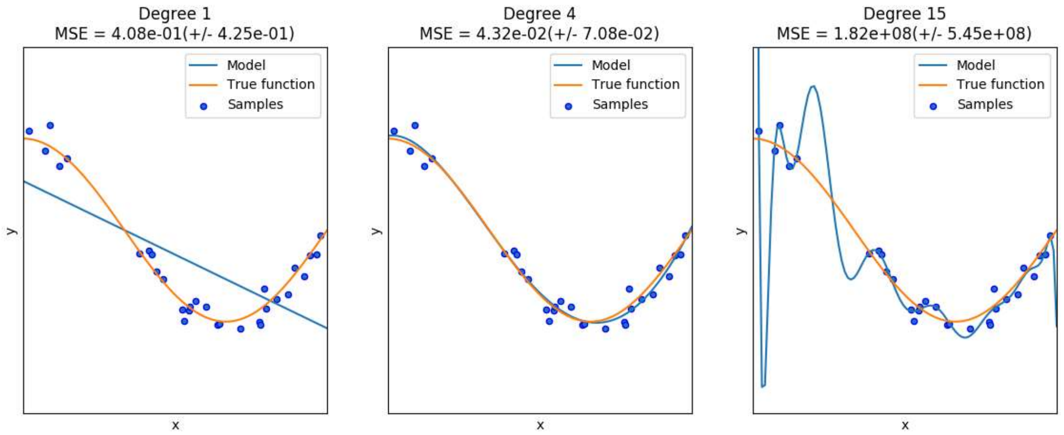
Very good observation! Increasing training data size does little help if the the testing score is already reached its optimum state. One main difference between traditional ML algorithms (statistical learning) and deep learning is that the latter usually can make use of much more data (Big Data) to improve its performance. Traditional ML algorithms tend to become plateau after passing a certain size of training data.



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.

Very good understanding of bias and variance.

In terms of overfitting and underfitting with different model complexity, sklearn also has a wonderful visualization to help us understand:



http://scikit-learn.org/stable/auto_examples/model_selection/plot_underfitting_overfitting.html

A more in depth explanation of bias and variance can be found at https://www.coursera.org/learn/ml-regression/home/week/3

You can audit it for free.

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

Evaluating Model Performance

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

You have a good understanding of what GridSearch is doing. But here you need to elaborate more on:

- Will it try out exhaustively all the combinations of all possible parameters of an algorithm or the combinations of parameters given by the engineer?
- How does it know which performs the best? Do we need to define an evaluation function?

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.

Very good start. To perfect your answer, can you elaborate more on:

- “splitting our original data into K-buckets” is not correct. Which data set will be split by GridSearchCV, the whole dataset or only the training data set (Hint: `reg = fit_model(X_train, y_train)`)?
- If we use the default Kfold method, the data will be split randomly or sequentially?
- Why K-fold cross validation is helpful to grid search? For example we could perform grid search without cross validation by splitting the training data into 8:2 training/validation set and train each parameter combination on the training set and get its score on the validation set to find the best. Besides less training data, what are the main drawbacks of grid search which are hinged upon using this particular method?

Student correctly implements the `fit_model` function in code.

Perfect implementation of `GridSearchCV`. Well done!

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

Note that the best max_depth returned here varies with different `random_state` in `train_test_split`, `ShuffleSplit` and `DecisionTreeRegressor`. You need to set them all in order to get a consistent result.

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.

You did a pretty good job in evaluating whether these prices are reasonable. Here we can compare their relative prices based on different features; compare their predicted prices with the real prices of houses with similar features and finally check whether their prices are within the range of max and min prices of the dataset.

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

Very good analysis. Can't agree with you more.

RESUBMIT

DOWNLOAD PROJECT

Learn the best practices for revising and resubmitting your project.

RETURN TO PATH

Student FAQ