UDACITY

PROJECT

Predicting Boston Housing Prices

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW

CODE REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Meets Specifications

Hello Udacian,
It was a great pleasure to me to review such great job. It is truly exceptional that you passed all the questions right in your first submission. Congratulation. You made it. ☺

Carry on your Great Work!

Data Exploration

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

Great job briefly exploring the Housing dataset and calculating it's key statistics! You also made good use of NumPy functionalities to calculate your results.

Pro Tips:

- Checking your dataset statistics is an very useful routine in applying a predictive model. This is because:
  - It helps us to check if the key assumptions of our algorithms hold (thereby helping us choose which model to apply).
  - These statistics tend to be very handy when you obtain a prediction, to check whether the predictions are reasonable, and not off-chart, compared to central values of the dataset.
- NumPy as a library might have been new for you, and not that easy to learn. In this tips section I'll give you some tips you can use when learning and picking up a new library:
  - Two functions are very useful when investigating a module (library) or a simple Python Object: the **doc** functionality and the dir() functionality.
  - If you wish to rapidly explore documentation of a library/module/function/object, you can just type print obj.**doc**, and the documentation of the function will be printed.
  - If you wish to rapidly explore what attributes and functions are available for an object, you can just type dir(obj), and you'll get a Python `list` of the object's attributes and functions.
  - Remember to always read documentation and try examples in your interpreter if you feel confused about a new library.
  - Hopes these help in your future Machine Learning Endeavors!
- Utilizing numPy is quite common when we are doing some statistical analysis no matter we are doing machine learning or data analysis. Thus, it is always useful for us to learn more the function inside numPy. Here is a course in udacity actually teaching us how to use it : intro to data analysis watch it if you want to learn more about it.
- Here is a website which provide lots of example work about most common used numpy function

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

Excellent work! You clearly understand the relationship between RM, LSTAT parameters with the house price.

Suggestions and Comments:

I do not agree that an increase in PTRATIO will lead to an increase in the value of the MDEV. Note that PTRATIO is the ratio of pupils to teachers. Let me explain why the answer is incorrect:

Imagine you have a kid. Will you like him/her to go to a school where one teacher takes care of 300 students, or a school where one teacher takes care of 5 students? I'm sure you'll prefer the latter. This should guide you in deciding how PTRATIO affects the price of the house (MDEV).

You did well explaining the features. We can always verify this by plotting the corresponding features against one another along with the line of best fit:

```
for feature in ["RM","LSTAT","PTRATIO"]:
#Scatter plot.
ax = data.plot(kind="scatter", x=feature,y="MEDV")

#Line of best fit (polynomial of degree 1).
w1, w0 = np.polyfit(data[feature],data["MEDV"],deg=1)

#Plot line of best fit.
rng = np.arange(np.min(data[feature]),np.max(data[feature]),0.1)
ax.plot(rng,[w0+w1*x for x in rng],color="red")
```

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.**

Great, you successfully executed the `R^2` score function and your discussion here is great!

Suggestions and Comments:

- Here is a useful website that helps me a lot to understand different performance metric. Check it out.

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.

You gave great reasons for splitting a dataset, and nice implementation using sklearn's `train_test_split`!

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.

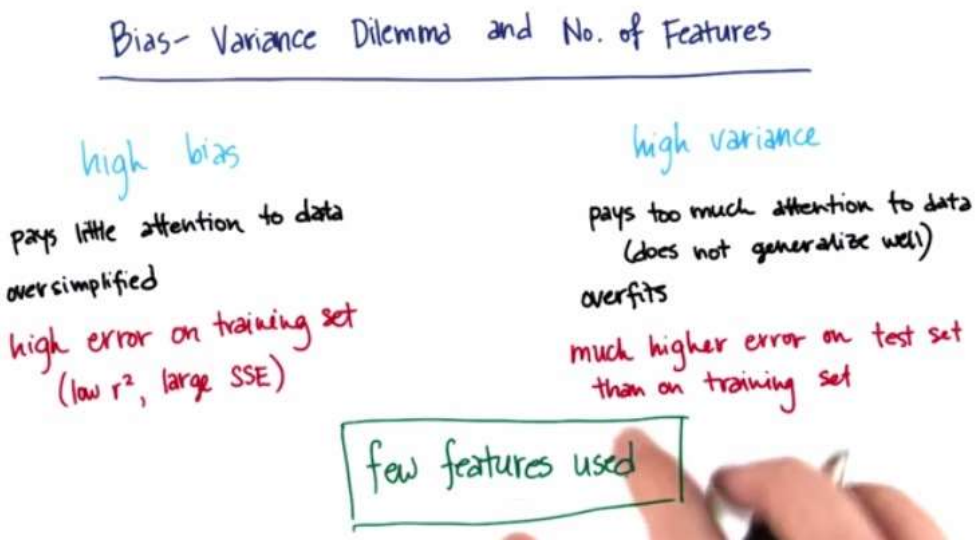
- Amazing job describing how the training and testing score change as the training set size increases.
- Adding more points is not beneficial. You're right! 😎

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.

Great job identifying that the model suffers from high bias when `max_dept = 1` and that the model suffers from overfitting (high variance) when the `max_dept = 10`.

Suggestions and Comments:

- Please check out this link if you want to understand more about model bias-variance tradeoff.



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

Nice rationale in guessing the best-guess optimal max depth for your work! 😄

Evaluating Model Performance

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

Awesome explanation of the grid search algorithm!

Suggestions and Comments:

- If you haven't done so yet, you may check out the scikit-learn page for GridSearchCV. This page gives an excellent explanation of GridSearchCV, which can be useful in fully grasping concepts underlining GridSearchCV.

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.

- Amazing description of k-fold cross validation and how it is performed on a model!
- You did an amazing job talking about how k-fold cross-validation is useful in grid search!

Student correctly implements the `fit_model` function in code.

Great implementation here!

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

Great job here! Your `max_depth` matched exactly the best-guess optimal model `max_depth` you gave earlier.

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.

Well done predicting, these are valid prices for your clients' houses. And also, your discussions on whether these prices are reasonable or not, sound great. 👍

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

- You made a very thorough and logical discussion here!

DOWNLOAD PROJECT

RETURN TO PATH

Student FAQ