UDACITY

< Return to "Machine Learning Engineer Nanodegree" in the classroom

DISCUSS ON STUDENT HUB

Predicting Boston Housing Prices

REVIEW

CODE REVIEW

HISTORY

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.

Good job the statistics for the Boston Housing dataset are accurately calculated.
Here is a performance comparison between numpy and pandas

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

You have correctly mentioned how each feature correlates with an increase or decrease in the target variable. Please have a look at the below image to see how correlation heatmap can be used to support our findings.

In my opinion:

1) Increasing the average number of rooms(RM) among homes in the neighborhood would increase the MDEV, since homes with more rooms have larger square footages which should raise the overal value of the home. This can be seen in the correlation heatmap too, the correlation between RM and MEDV is 0.70 ie strongly correlated.

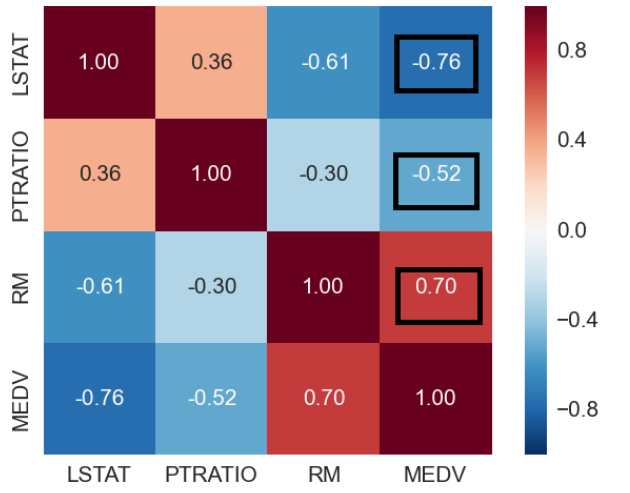
2) Increasing the percentage of "lower class" homeowners in the neighborhood (LSTAT) should decrease the MDEV, as this could mean that a particular neighborhood is located at an area where lower income/class people live and these people are probably only able to afford lower cost homes. This is confirmed from the correlation value of -0.76 that means there is a strong negative correlation between LSTAT and MEDV.

3) Increasing the ratio of students to teachers (PTRATIO) would lead to a decrease in MEDV. In general, better school have lower teacher to student ratios, as allows for more access/interaction between a given teacher and a smaller group of students. That being said, most families look for homes in located in neighborhoods with good schools for the benefit of their kids' education, which drives up the prices of the homes in those neighborhoods. Therefore, assuming that increasing the PTRATIO correlates to worse schools in the neighborhood, then the prices of the homes (MEDV) should decrease. This is reflected with a negative correlation value of -0.52 that shows an inverse relation between PTRATIO and MEDV

Seaborn Correlation Heatmap to understand feature correlation with MEDV

```
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

cols = ['LSTAT', 'PTRATIO', 'RM', 'MEDV']
cm = np.corrcoef(data[cols].values.T)
sns.set(font_scale=1.5)
hm = sns.heatmap(cm,
                 cbar=True,
                 annot=True,
                 square=True,
                 fmt='.2f',
                 annot_kws={'size': 15},
                 yticklabels=cols,
                 xticklabels=cols)
plt.show()
```



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.

The performance metric is correctly implemented and the discussion on how successfully the model captures the variation of the target variable is correct. Here is some more information on R-squared.

Statisticians and scientists often have a requirement to investigate the relationship between two variables, commonly called x and y. The purpose of testing any two such variables is usually to see if there is some link between them, known as a correlation in science. For example, a scientist might want to know if hours of sun exposure can be linked to rates of skin cancer. To mathematically describe the strength of a correlation between two variables, such investigators often use R2.

Linear Regression

Statisticians use the technique of linear regression to find the straight line that best fits a series of x and y data pairs. They do this through a series of calculations which derive the equation of the best line. This mathematical description of the line will be a linear equation and have the general form of y = mx + b, where x and y are the two variables in the data pairs, m is the slope of the line and b is its y intercept.

Correlation Coefficient

The calculations which find the best straight line will produce a linear equation to fit any set of data, even if that data is not actually very linear. In order to have an indication of how well the data actually fit a straight line, statisticians also calculate a number known as the correlation coefficient. This is given the symbol r or R and is a measure of how closely aligned the data pairs are to the best straight line through them.

Significance of R

R can have any value between -1 and 1. A negative value of R simply means that the best fit straight line slants downwards moving left to right, rather than upwards. The closer R is to either the of the two extremes, the better the fit of the data points to the line, with either -1 or 1 being a perfect fit and an R value of zero meaning that there is no fit and the points are totally random. If the data points are well aligned to the straight line, there is said to be some correlation between them, hence the name correlation coefficient for R.

R2

Some statisticians prefer to work with the value of R2, which is simply the correlation coefficient squared, or multiplied by itself, and is known as the coefficient of determination. R2 is very similar to R and also describes the correlation between the two variables, however it is also slightly different. It measures the percent of variation in the y variable which can be attributed to variation in the x variable. An R2 value of 0.9, for example, means that 90 percent of the variation in the y data is due to variation in the x data. This does not necessarily mean that x is truly affecting y, but that it appears to be doing so.

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. Here is some more information on how train, validation and test sets work:

Training Dataset

> **Training Dataset**: The sample of data used to fit the model.

The actual dataset that we use to train the model (weights and biases in the case of Neural Networks). The model *sees* and *learns* from this data.

Validation Dataset

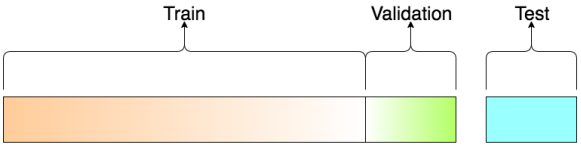
> **Validation Dataset**: The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters. The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration.

The validation set is used to evaluate a given model, but this is for frequent evaluation. We as machine learning engineers use this data to fine-tune the model hyperparameters. Hence the model occasionally *sees* this data, but never does it "*Learn*" from this. We(mostly humans, at-least as of 2017 😛) use the validation set results and update higher level hyperparameters. So the validation set in a way affects a model, but indirectly.

Test Dataset

> **Test Dataset**: The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset.

The Test dataset provides the gold standard used to evaluate the model. It is only used once a model is completely trained(using the train and validation sets). The Test set is generally what is used to evaluate competing models (For example on many Kaggle competitions, the validation set is released initially along with the training set and the actual Test set is only released when the competition is about to close, and it is the result of the the model on the Test set that decides the winner). Many a times the validation set and the Test set serve is used as the Test set, but it is not good practice. The Test set is generally well curated. It contains carefully sampled data that spans the various classes that the model would face, when used in the real world.

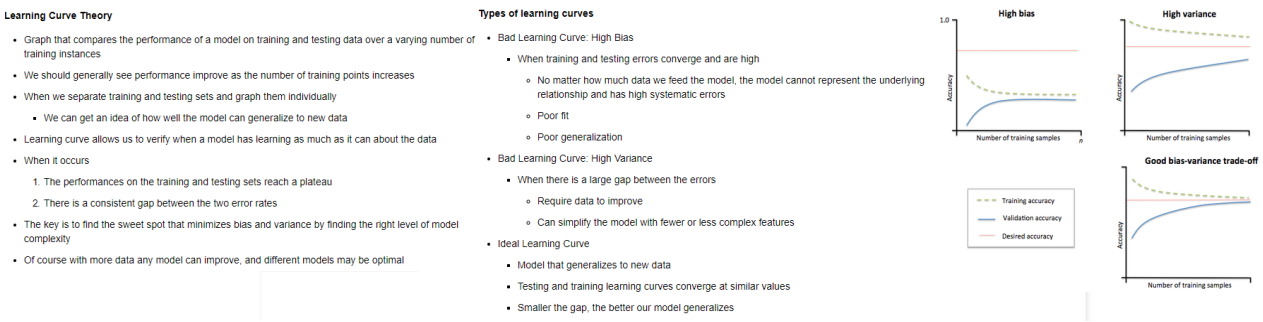


Remember, the test set is data you don't touch until you're happy with your model. The test set is used only **ONE** time to see how your model will generalize. That's it.

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.

Good job on the discussion, we see that training score decreases, testing score increases (and possibly decreases depending on depth). Adding more points is not beneficial after a certain point.
Here is some more information on learning curves:



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.

As we can see a max_depth of 1 suffers from high bias; a max_depth of 10 suffers from high variance. Well done! the reasoning for the complexity curves is absolutely correct.
Here is some more information on how to identify models suffering from high bias/ high variance and what happens at max_depth = 1 and max_depth=10

Identifying when a model is suffering from high bias or high variance.

- It is easy to identify whether the model is suffering from a high bias or a high variance.
  - High variance models have a gap between the training and validation scores.
    - This is because it is able to fit the model well but unable to generalize well resulting in a high training score but low validation score.
  - High bias models have have a small or no gap between the training and validations scores.
    - This is because it is unable to fit the model well and unable to generalize well resulting in both scores converging to a similar low score.

Maximum depth of 1: High Bias

- Both training and testing scores are low.
- There is barely a gap between the training and testing scores.
- This indicates the model is not fitting the dataset well and not generalizing well hence the model is suffering from high bias.

Maximum depth of 10: High Variance

- Training score is high. Testing score is low
- There is a substantial gap between the training and testing scores.
- This indicates the model is fitting the dataset well but not generalizing well hence the model is suffering from high variance.

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

Good job selecting the best guess optimal model 👍🏽

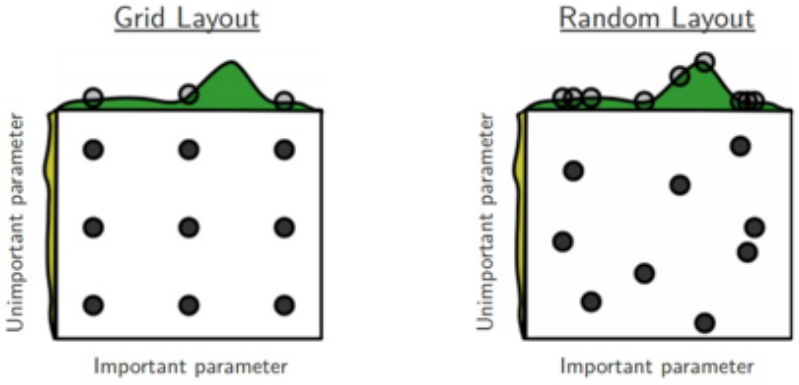
Evaluating Model Performance

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

Great job on clearly understanding this technique.
Here is some more information on grid search technique:

In the "grid search" process we define a set of parameter values that we want to try with a given model, and then we use cross-validation to evaluate every possible combination of those values in order to choose between them.

However, GridSearchCV can be computationally expensive. For example, searching 10 different parameter values for each of four parameters will require 10,000 trials of cross-validation, which equates to 100,000 model fits and 100,000 sets of predictions if 10-fold cross-validation is being used. One solution is to do a "random search" instead, using RandomizedSearchCV:



In a random search process, we search only a random subset of the provided parameter values. This allows us to explicitly control the number of different parameter combinations that are attempted, which we can alter depending on the computational time you have available.

It's certainly possible that RandomizedSearchCV will not find as good a result as GridSearchCV, but we might be surprised how often it finds the best result (or something very close) in a fraction of the time that GridSearchCV would have taken. And when given the same computational budget, RandomizedSearchCV can sometimes outperform GridSearchCV when continous parameters are being searched, since a random search process leads to a more fine-grained search.

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

Here is some more information on KFold cross validation:

Cross-validation is a technique to evaluate predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it.

In k-fold cross-validation, the original sample is randomly partitioned into k equal size subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k-1 subsamples are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data. The k results from the folds can then be averaged (or otherwise combined) to produce a single estimation. The advantage of this method is that all observations are used for both training and validation, and each observation is used for validation exactly once.

For classification problems, one typically uses stratified k-fold cross-validation, in which the folds are selected so that each fold contains roughly the same proportions of class labels.

In repeated cross-validation, the cross-validation procedure is repeated n times, yielding n random partitions of the original sample. The n results are again averaged (or otherwise combined) to produce a single estimation.

- K-fold cross-validation summary:
  - Dataset is split into K "folds" of equal size.
  - Each fold acts as the testing set 1 time, and acts as the training set K-1 times.
  - Average testing performance is used as the estimate of out-of-sample performance.
  - Also known as cross-validated performance.
- Benefits of k-fold cross-validation:
  - More reliable estimate of out-of-sample performance than train/test split.
  - Reduce the variance of a single trial of a train/test split.
- Hence, with the benefits of k-fold cross-validation, we're able to use the average testing accuracy as a benchmark to decide which is the most optimal set of parameters for the learning algorithm.
  - If we do not use a cross-validation set and we run grid-search, we would have different sets of optimal parameters due to the fact that without a cross-validation set, the estimate of out-of-sample performance would have a high variance.
  - In summary, without k-fold cross-validation the risk is higher that grid search will select hyper-parameter value combinations that perform very well on a specific train-test split but poorly otherwise.

Limitation of k-fold cross-validation:

- It does not work well when data is not uniformly distributed (e.g. sorted data).

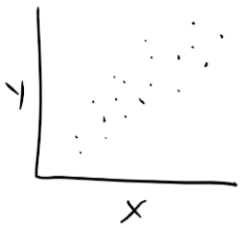
Student correctly implements the `fit_model` function in code.

The fit_model has been correctly implemented 👍

Here is some more information on what fitting a model means:

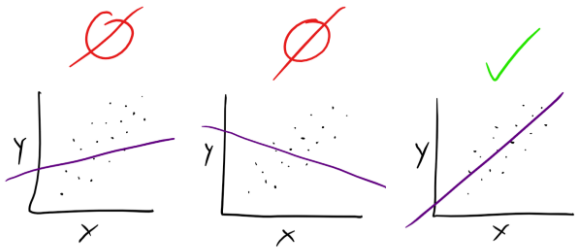
What on earth do people mean when they tell you they're going to "fit a model"?

Let's start with what a model is. A model is a description of a system, usually expressed as an equation of some kind. Let's say we have some data – measurements of variables **x** and **y**. We think that in the future, we'll have measurements of more **x**-like values, and we'd like to be able to predict those **y**s. It looks like you could draw a nice straight line through this cloud of points. That means that a linear model might be a good choice for this data. We've just done the first step in the model-fitting process: we've decided to use a line – a simple linear model.



Some "data".

The process of picking the correct line for this model is called "fitting". There are different ways to do this – least squares is possibly the most familiar one. You could also use the "wiggle a ruler around on paper" method, or the "draw lines in Powerpoint" method. We'll skip the details of that step because the internet describes least-squares fairly well.



Two bad fits and one good one.

I'm going to use an equation here, but it's simple and it's the only one I'll use in this post!

That fitted line can be described with the equation **y=mx+b.** When we fit the model what we're really doing is choosing the values for **m** and **b** – the slope and the intercept. The point of fitting the model is to find this equation – to find the values of **m** and **b** such that **y=mx+b** describes a line that fits our observed data well. In the case of the best fit model above, **m** is close to 1, and **b** is just a bit larger than 0.

And why do we care about this? Well, that value of **m** can be really informative. If **m** is very large and positive, then a small change in the value of **x** predicts a tremendous positive change in the value of **y.** If **m** is small, then changes in **x** predict small changes in **y.** A large **m** implies that **x** may have a large *effect* on **y**, hence m is also sometimes called the *effect size.* It's also sometimes called a *coefficient.*

The principals of this model-fitting can be applied to linear models created on data with many more parameters – many dimensions. But they are fit in similar ways. We may not be able to draw multidimensional datasets in neat graphs but we can still apply least-squares to them!

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

Good job the optimal model is correctly chosen

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.

Your predictions do seem reasonable.
When I ran this cell I got the following predictions, just sharing my observations hope you dont mind

Prices to recommend

- Client 1: 406,900
- Client 2: 232,200
- Client 3: 938,100

Data Exploration Findings

- Minimum price: $105,000.00
- Maximum price: $1,024,800.00
- Mean price: $454,342.94
- Median price $438,900.00
- Standard deviation of prices: $165,340.28

Reasonableness

- The prices are rounded up to the nearest hundred as the prices in the dataset are all rounded to the nearest hundred.
- Compared to the data-exploration, it seems that the houses' prices from client 1 and client 2 are below the mean and median prices.
  - For client 2, it seems reasonable due to the high poverty level and student-to-teacher ratio.
  - For client 1, it also seems reasonable due to the average poverty level and student-to-teacher ratio.
- And the house's price from client 3 is way above the mean and median prices, nearing to the maximum price in the dataset.
  - This seems reasonable given the low poverty level and student-to-teacher ratio with a high number of rooms.

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

Good job on the discussion 
Here was my argument:

This model is hard to use in the real life.

1. Today the data from 1978 (even multiplicatively scaled to account for 35 years of market inflation) is not relevant anymore because the real estate market as well as population, salaries, demand have changed dramatically with the buyers.
2. How people choose the house or the features of the dataset are not sufficient anymore. People just want something different from the house (is there WiFi inside? is it close to my corporation? etc.). Adding meaningful features to our model may be a good way of reducing the so-called irreducible error, or the error that comes from variation in our target that our predictors are not able to explain.
3. The model is not robust enough to make consistent predictions. As was shown in the Sensitivity section, range in prices is around $70k which might be significant.
4. More examples are needed: from the different types/sizes of cities, countries, etc.

At the same time there is so much data generated in the World nowadays and the usage of the right features let us build the robust applicable models for solving real-world problems.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review