Kenytt Avery, Computer Science Department, California State University, Fullerton

CPSC 483 - Introduction to Machine Learning

Project 3, Spring 2021

due March 25 (Section 01) / March 26 (Section 02)

Last updated Thursday, March 11, 4:15 pm PST

In this project you will use scikit-learn, which is a higher-level machine learning library that works with NumPy, to compare the performance of linear and polynomial regression models.

The project may be completed individually, or in a group of no more than three (3) students. All students on the team must be enrolled in the same section of the course.

Platforms

The platform requirements for this project are the same as for Project 1.

Libraries and Code

In addition to NumPy and pyplot, you will need scikit-learn. You may not use any other library except the Python Standard Library.

Code from *A Whirlwind Tour of Python*, the Jupyter notebooks accompanying the textbook, and from the library documentation may be reused. All other code and the results of experiments must be your own original work or the original work of other members of your team.

Dataset

This project uses the same `boston.npz` file as Project 2.

Experiments

Run the following experiments in a Jupyter notebook, performing actions in code cells and reporting results in Markdown cells.

1. Use the NumPy `load()` method to read the dataset. The data contains two arrays: `'features'`, which contains the variables CRIM through LSTAT, and `'target'`, which contains the variable MEDV.

2. Use `sklearn.model_selection.train_test_split()` to split the features and target values into separate training and test sets. Use 80% of the original data as a training set, and 20% for testing. To make sure that your results are reproducible, pass `random_state=(2021-3-11)`.

3. Create a scatterplot of the training data showing the relationship between the percentage of the population that is lower status and the median value of a home. Does the relationship appear to be linear?

   (Note that "status" here refers to socioeconomic status and is *not* a value judgement on the residents.)

4. With LSTAT as ***X*** and MEDV as ***y***, create and `fit()` an `sklearn.linear_model.LinearRegression` model. Using the `coef_` and `intercept_` attributes of the model, what is the equation for MEDV as a linear function of LSTAT?

5. Use the `coef_` and `intercept_` attributes of the model to add a line representing the least squares fit to your scatter plot from experiment *(3)*. How well does the model appear to fit the training data?

6. Use the `predict()` method of the model to find the response for each value of the LSTAT attribute in the training set. Using `sklearn.metrics.mean_squared_error()`, find the average loss $\mathcal{L}$ for the training set.

7. Repeat experiment *(6)* for the test set. How do the training and test MSE values compare?

8. Let's see if we can fit the data better with a more flexible model. Use `np.hstack()` to add a degree-2 polynomial feature to ***X***, then fit a new linear model. How do the training and test MSE values for this model compare to the previous model?

9. Repeat experiment *(5)* for your polynomial model.

10. Repeat experiments *(4)*, *(6)*, and *(7)* using all 13 input features as ***X***. How do the training and test MSEs for this model (which is a linear model including all features) compare to the values you found for experiment *(8)* (which was a degree-2 polynomial model including a single feature)? What accounts for the difference?

11. Combine experiments *(8)* and *(10)*, using `np.hstack()` to add the squares of all 13 input features to ***X***. How do this model's training and test MSE scores compare to the previous model using all 13 features?

12. Scikit-learn is also capable of constructing polynomial features for us using `sklearn.preprocessing.PolynomialFeatures`, but those features also include interaction features, where the feature terms are multiplied together.

Use the `fit_transform()` method to create degree-2 polynomial and interaction terms for the original set of 13 features, then fit a new linear model. Compare the training and test MSE to the previous model. What is the effect of adding interaction terms in this case? Do we seem to be overfitting?

Submission

A Markdown cell at the top of the notebook should include project summary information as described in the Syllabus for README files.

Since you may be actively editing and making changes to the code cells in your notebook, be certain that each of your code cells still runs correctly before submission. You may wish to do this by selecting *Run All* from the drop-down menu bar.

Submit your Jupyter `.ipynb` notebook file through Canvas before class on the due date.

If the assignment is completed by a team, only one submission is required. Be certain to identify the names of both students at the top of the notebook. See the following sections of the Canvas documentation for instructions on group submission:

- How do I join a group as a student?
- How do I submit an assignment on behalf of a group?