# Machine learning course assignments

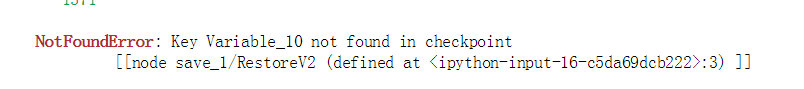
2/6/2020

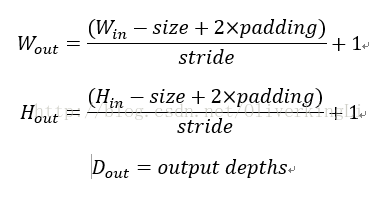
1. **Boston house price forecast**

Peter asked for two different ways to achieve the Boston house price forecast. The first method is the commonly used method of finding the best "max-depth" parameter through continuous training and decision tree algorithm, which lays the foundation for the final prediction. After searching the corresponding code on the Internet, this method can be realized with a fitting degree of 0.923. However, there are still some problems when running on Jupyter. For example, some function names in Python3 have been changed and the latest function names need to be adopted. During the visual execution, the program kept reporting errors, looked for solutions on the Internet, and tried several methods, but failed. Finally before the report, with the help of the teacher to find a solution to the problem.

Peter introduced many machine learning algorithms at the beginning of the course, including KNN, DecisionTrees, ANN, SVR and so on. Therefore, KNN algorithm was selected to try to solve the Boston housing price prediction problem. Access to information, import the required libraries first, and then the data is normalized processing, using KNN model to predict prices, found as n\_neighbors increases gradually, gradually enhance its capability of models to predict, but when n\_neighbos greater than 2 model score tends to decline. So let's say n\_neighbors=2.According to the test results, the score of KNN model is 0.693957.At the same time, I tried SVR, but the result was obviously incorrect. I still need to learn and improve. The process is described in the document "波士顿房价预测作业2".

1. **CNN – Tensorflow to implement LeNet – 5**

In the course, I did not understand the principle of CNN, so I consulted relevant explanations of CNN on CSDN after class. For the requirements of the project2. I want to try with the dimension of the convolution kernels for 3 x3 convolution, the problem is that after the convolution kernel size changes, the number of the corresponding number of convolution kernels and featuremaps has changed too, although I have found the convolution calculation formula and pooling layer calculation formula: , according to the formula, the parameters of each layer under the convolution are changed accordingly, but the program is running into the model training error, can not find a solution. There has always been some confusion about how to determine the number of convolution kernels in different convolutional layers. The number of convolution kernels in the first layer is 6, which becomes 16 in the second convolution operation. Finally, decided to annotate the core code in LeNet-lab-solution, which was specifically annotated in " LeNet-Lab-Solution ".



1. **Summary**

This machine learning course is my first contact with python and many algorithms, and I learned to use GitHub and solve simple code errors. I will continue to learn relevant knowledge in the future to deepen my understanding of machine learning. Finally, thanks to Peter for four days of teaching.

--M105119301 方铖

[Jupyter Notebook](http://ipython.org/notebook.html)