# CPSC 483 - Introduction to Machine Learning

Project 3, Fall 2020

due October 26 (Section 02) / October 29 (Section 01)

*Last updated Saturday October 10, 10:30 pm PDT*

Having looked “under the hood” at NumPy implementations of linear and polynomial regression in [Project 2](https://docs.google.com/document/d/1ZYSWc1Daq72-38QKeVJmR-ovNvKf3QTEPcs8A8OrSc8/edit?usp=sharing), we return to scikit-learn to see what it can offer in terms of automating cross-validation procedures.

The project may be completed individually or in a group of no more than three (3) people. 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](https://docs.google.com/document/d/1gAvnkp62x00YTYp9Vwo0qbuQQqpxY9pGykYewmgAMAg/edit?usp=sharing) and Project 2.

## Libraries

You will need [scikit-learn](https://scikit-learn.org/) to obtain the data, build models, and run cross-validation. You may also wish to use [pandas](https://pandas.pydata.org/) DataFrames to examine and work with the data, but this is not a requirement.

You may reuse code from the [Jupyter notebooks accompanying the textbook](https://github.com/sdrogers/fcmlcode/tree/master/notebooks) and from the documentation for the libraries. All other code and the results of experiments should be your own.

## Dataset

The scikit-learn [sklearn.datasets](http://scikit-learn.org/stable/datasets/index.html) module includes some small datasets for experimentation. In this project we will turn the tables on the [Boston house prices dataset](https://scikit-learn.org/stable/datasets/index.html#boston-house-prices-dataset). The original use of the dataset was to try and predict the median value of a home given several features of its neighborhood. One of features, however, is CRIM, the per-capita crime rate by town.

In this project we will engage in some amateur [predictive policing](https://www.brennancenter.org/our-work/research-reports/predictive-policing-explained), attempting to predict the crime rate using the other features in the dataset.

See the section on [scikit-learn](https://kolesnikov.ga/Datasets_in_Python/#scikit-learn) in Sergiy Kolesnikov’s blog article [Datasets in Python](https://kolesnikov.ga/Datasets_in_Python/) to see how to load this dataset and examine it using pandas DataFrames.

## Experiments

Run the following experiments in a Jupyter notebook, performing each action in a [code cell](https://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/Running%20Code.html) and answering each question in a [Markdown cell](https://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/Working%20With%20Markdown%20Cells.html).

1. Load and examine the Boston dataset’s features, target values, and description.
2. Save CRIM as the new target value ***t***, and drop the column CRIM from ***X***. Add the target value MEDV to ***X***.
3. Use [sklearn.model\_selection.train\_test\_split()](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html) 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.
4. Create and [fit()](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html#sklearn.linear_model.LinearRegression.fit) an [sklearn.linear\_model.LinearRegression](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html) to the training set.
5. Use the [predict()](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html#sklearn.linear_model.LinearRegression.predict) method of the model to find the response for each value in the test set, and [sklearn.metrics.mean\_squared\_error()](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html), to find the training and test MSE.
6. By itself, the MSE doesn’t tell us much. Use the [score()](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html#sklearn.linear_model.LinearRegression.score) method of the model to find the *R2* values for the training and test data.

*R2*, the *coefficient of determination*, measures the proportion of variability in the target ***t*** that can be explained using the features in ***X***. A value near 1 indicates that most of the variability in the response has been explained by the regression, while a value near 0 indicates that the regression does not explain much of the variability. See Section 3.1.3 of *An Introduction to Statistical Learning* for details.

Given the *R2* scores, how well did our model do?

1. Let’s see if we can fit the data better with a more flexible model. Scikit-learn can [construct polynomial features](https://scikit-learn.org/stable/modules/linear_model.html#polynomial-regression-extending-linear-models-with-basis-functions) for us using [sklearn.preprocessing.PolynomialFeatures](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PolynomialFeatures.html) (though note that this includes interaction features as well; you saw in Project 2 that purely polynomial features can easily be constructed using [numpy.hstack()](https://numpy.org/doc/stable/reference/generated/numpy.hstack.html)).

Add degree-2 polynomial features, then fit a new linear model. Compare the training and test MSE and *R2* scores. Do we seem to be overfitting?

1. Regularization would allow us to construct a model of intermediate complexity by penalizing large values for the coefficients. Scikit-learn provides this as [sklearn.linear\_model.Ridge](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html). The parameter alpha corresponds to 𝜆 as shown in the textbook. For now, leave it set to the default value of 1.0, and fit the model to the degree-2 polynomial features. Don’t forget to normalize your features.

Once again, compare the training and test MSE and *R2* scores. Is this model an improvement?

1. We used the default penalty value of 1.0 in the previous experiment, but there’s no reason to believe that this is optimal. Use [sklearn.linear\_model.RidgeCV](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RidgeCV.html) to find an optimal value for alpha. How does this compare to experiment *(8)*?

## Submission

Submit your Jupyter .ipynb notebook file through Canvas before class on the due date. Your notebook should include the usual identifying information found in a README.TXT file.

If the assignment is completed by a team, only one submission is required. Be certain to identify the names of all students on your team at the top of the notebook.