Classes 2 and 3: Machine learning workflow and regression machine learning

Professor Michael Dowling, Dublin City University

Purpose of these classes

We're going to start (in Class 2) by going through how to set up machine learning (ML) from start to finish. This involves the core element of 'working with data', as it became clear from some feedback after the last class that you might not have the knowledge of this that I was assuming you had.

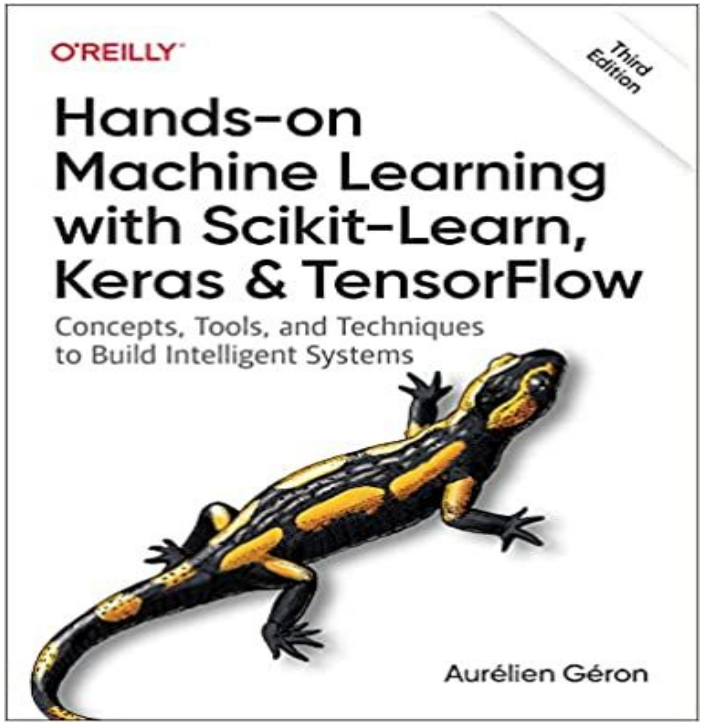
We're then going to look at linear regression (Class 3) as a core ML technique. A method you are all familiar with, but applied in a ML style. We'll also look at how to make standard linear regression more advanced to take full advantage of the methods. This includes Support Vector Machines, constrained regressions, and ensemble models.

After these two classes you should be comfortable taking a raw dataset, turning it into the format you need for analysis, and then running a reasonably advanced ML linear regression to discover relationships.

Reading support for today

It’s definitely worth reading Chapter 2 (data analysis) of this course text. Other reading includes Chapters 4, 5, 6, 7 (just a skim-read)

This material is primarily learning-by-doing, so lets open up Google Colab straightaway.



Concepts raised in the worksheet

Root Mean Square Error

Root Mean Square Error (RMSE) is the standard way to measure the performance of regression-style ML models, essentially a measure of how much errors are made in predictions compared to actual values. Large errors are particularly penalised.

If you know there are quite a few outliers in the data and you don't want to penalise them excessively an alternative method is **Mean Absolute Error (MAE)**, but there should be a compelling reason to use this rather than the standard RMSE.

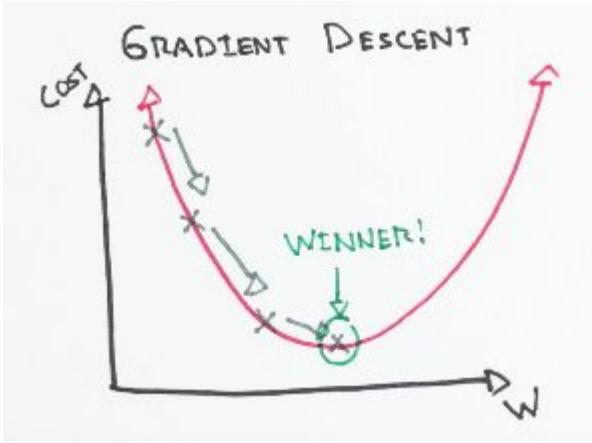
$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

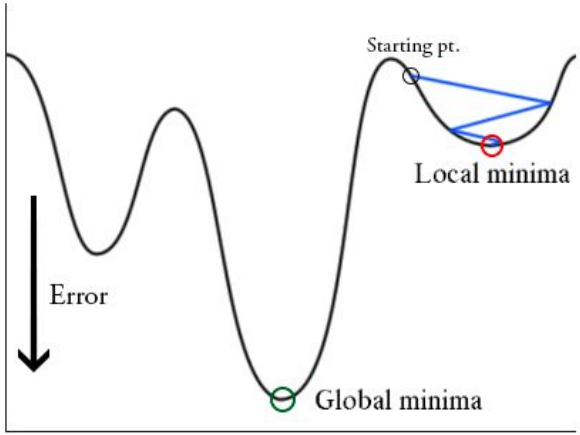
Gradient Descent

A popular algorithm to minimise RMSE through iteration.

A key to ‘good’ gradient descent implementation is to set the learning rate parameter correctly, so that you (a) arrive at the minimum value before all set iterations are used, and (b) you don’t overshoot the minimum value

There is also a risk of just finding a local minimum, rather than a global minimum, which can be overcome by setting enough iterations.





Decision Trees

Decision Trees can be a form of regression (continuous dependent variables) or classification (1, 0 dependent variables). We are applying decision trees in our analysis as a form of regression as our dependent variable is house prices.

As a regression, decision trees estimate multiple layers of relationships, by estimating regression equations for subsets of variables and building deeper understanding as we move down the decision tree.

Decision trees are a standard ML application and particularly suitable for largish numbers of independent variables where you need to distinguish variable importance.

An issue with decision trees is that they are highly sensitive to training data and, thus, prone to overfitting.

Random Forest regression

We'll look at this again a bit more later in the course, but for now random forests can be assumed to be an 'ensemble' (collection) of models built on random subsets and slices of the features of the model, with the produced result being an average of the results of these subset results.

A key advantage of random forests over a more basic model is that it enables better identification of a features importance. We might, for example, see that one variable performs very well on all data, but has less strength on subsets of data (maybe its importance is due to how it works with outliers, for example), while another variable consistently performs well across all types of subsets of data. The random forest method will allow boosting the importance of that latter variable.

Ridge regressions

Adds a constraint to regular linear regressions that seeks to not only fit the data to the model, but also seeks to keep the model as small as possible.

In practice this is done by the introduction of a hyperparameter to the model, alpha, which says how constrained the final model should be. An alpha value of 0 means there are no constraints, and therefore the model is the same as a regular linear regression. A high alpha value adds high burdens to variables being considered useful in the model due to very low weights attached to variables.

We normally run the constraint only on the training dataset, and then run the unconstrained model on the test dataset.

Lasso regression

An old econometric technique from the 70s that has become newly popular in machine learning

Similar to ridge regression in intent, a major difference though is that while ridge retains all variables, even if lowly weighted, lasso will eliminate variables in the modelling that have very low weights.

This is quite useful in terms of presenting a reduced variable model to best describe a dataset

Elastic Net regression

Elastic Net regression can be best considered as a halfway point between Ridge regression and Lasso regression. It can both assign low weights to variables and set certain variables to zero.

Elastic Net is preferred over Lasso when certain variables are strongly correlated, as Lasso might choose at random which of the highly correlated variable to set to zero. While Elastic Net might instead choose to retain variables but just set the weights low.

If unsure whether to use Ridge, Lasso, Elastic Net, it is best to go with Ridge as a default, then Elastic Net, then Lasso.

Support Vector Machine

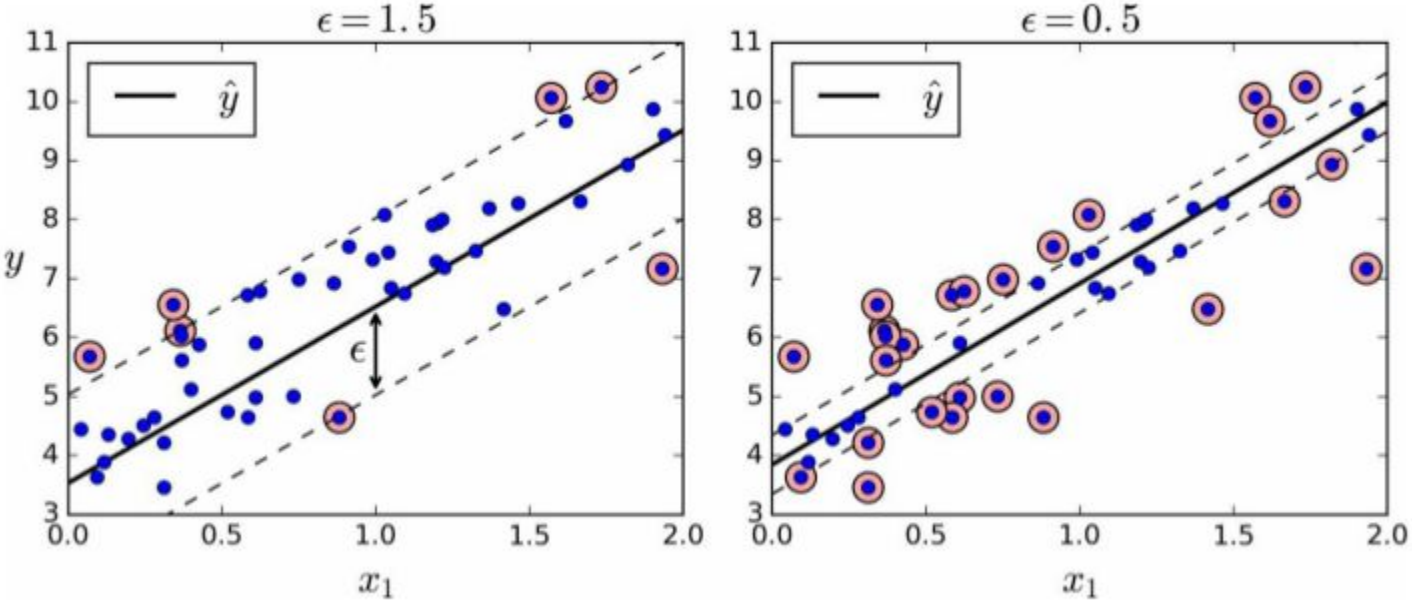


Figure 5-10. SVM Regression

SVM regression

SVM is primarily used for classification - through trying to identify separation between groups, however it can be inverted to identify regression relationships.

A hugely popular method, that should be used as a default part of any machine learning set of models.

SVM regression essentially tries to fit as many instances (predicted outcomes) as possible on a line, while minimising violations. Can adjust the width of the line (upper and lower bounds of the line) through hyperparameters.

Gradient Boosting

[Not to be confused with gradient descent]

Gradient boosting improves prediction by sequentially adding predictors to create an ensemble model. The new predictors are based on examination of the residual errors from a regression, attempting to find predictors based on patterns in these residual errors.

Gradient Boosting - graphically