# Introduction to Data Science

Ryan Greenup ; 1780-5315

November 17, 2019

# Contents

# (Wk 1) Introduction to Statistical Learning

*Topic 1* | *Chapters 1 & 2 of ISL* [?ISL]

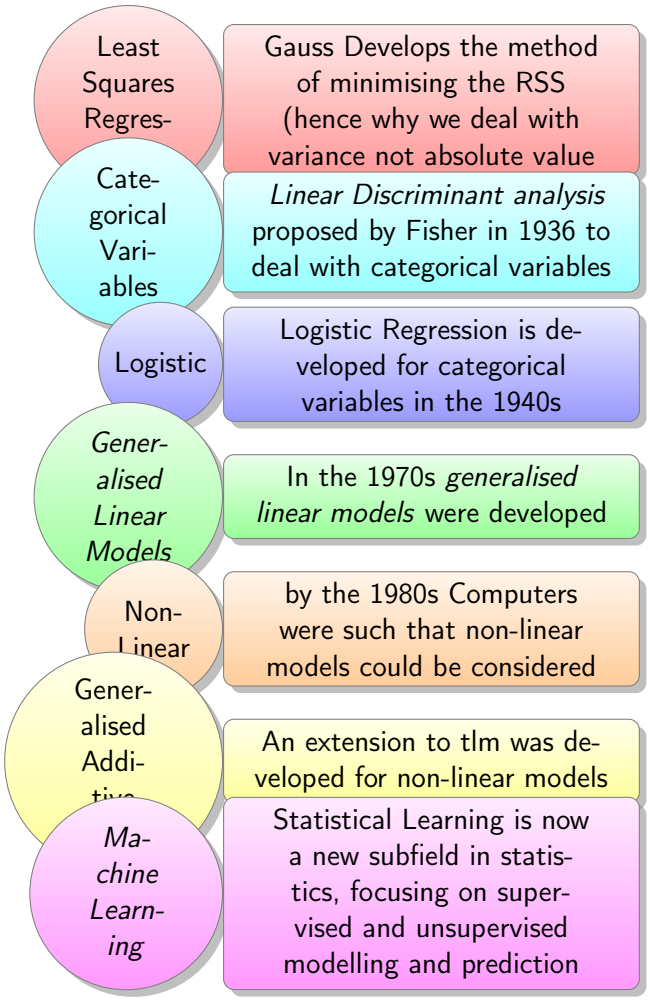
## Types of Learning

Generally there are two types of statistical learning, supervised and unsupervised:

- Supervised learning is concerned with creating a model to relate input data to ouput data
- UnSupervised is concerned with deciding upon what an output might be and hence creating a model to model as such
  - unsupervised basically means that no pre-determined output variable has been recognised, for example grouping flowers together by observed characteristics would be known as *clustering*, which is an unsupervised technique

## Statistical Learning Over Time

Much of the techniques of Statistical Learning can be better understood by considering how it developed over time:




# What is statistical Learning

Statistical Learning involves observing data and assuming that the observed data is of the form:

$$Y \; f\left(x\right) + \varepsilon \tag{1.1}$$

It is assumed that stochastic error is a fundamental property of our universe.

## Types of Error

When structuring a model it is worth bearing in mind the types of errors that will be inherit to the system:

- Deterministic Error

  - Overall Trend of the Data
  - Seasonal Trend $S_t$
    * A trend that oscillates with at a predictable frequency (like weather seasons)
  - Cyclical Trend
    * A trend that will occur but not with a predictable frequency, like boom/bust cycles or El Niño and La Niña

- Stochastic Error
  - Random Error
    * Unforseeable Events such as a sunspot effecting temperature.
  - Systemic Errors
    * The inability to measure something more accurately, e.g. rulers going only to mm, or the fundamental limit of Planck's length $1.6 \times 10^{-35}$

## Reasons to estimate a descriptive Function

The function that we have initally assumed to exist in (1.1) may be useful in two ways:

- Infering how the response variable behaves given changes in predictive values
- Forecasting what future behaviour will be given past behaviour [1]

Depending on our primary interest, this may effect the model we choose, for example, a simple linear regression may be less accurate than a really complicated model, however, it is really easy to interpret which variables ought to be manipulated in order to acheive a specific outcome. If however our primary interest was in predicting future values (e.g. depreciation of car value) then we would want the most accurate model regardless of how complicated that is.

## Parametric and Non-Parametric Methods

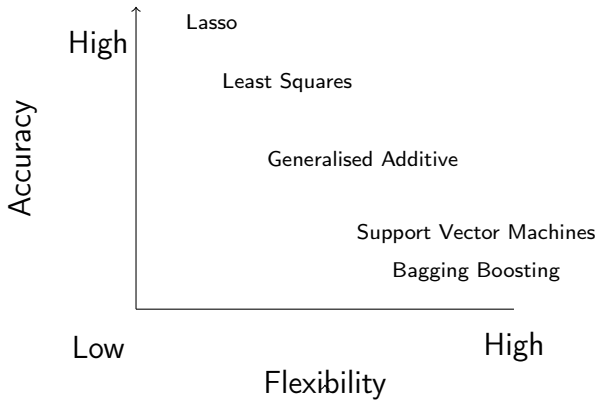
Parametric methods presume a mathematical function that describes the data and then simply solves for the parameters of the model.

A non-parametric method is basically a way of drawing a shape through the data that isn't to wobbly and is appropriately smooth (i.e. is not overfitted).

Non-parametric methods have the advantage that they are not restricted in accuracy by a potentially flawed underlying model, however they require far more data and are not-interpretable.

[1] ISL p. 17, [2.1.1] [?ISL]

Often Prediction accuracy and Model Interpretability is a trade off [2]:




So if we wanted a model that we could interpret a lot o infomation from, it may be more appropriate to choose a *Lasso* regression rather than a *Generalised Additive Model*, however if we wanted significantly more accurate predictions at the loss of interpetability perhaps a *Bagging-Boosting* method would be more appropriate. [3]

### Overfitting

More flexibile methods are also prone to overfitting, for example a linear regression will create more accurate predictions of the distance travelled by a particle of constant velocity, period. any other models will be taking into consideration noise.

## Supervised and Unsupervised Learning

Supervised learning is concerned with creating a model between input and output data, Unsupervised learning is basically what happens when there is no clear output data (e.g. clustering species together). [4]

## Regression and Classificatoin

Generally when an ouptut is categorical/discrete a modelling problem will be referred to as classification because given predictors, the ouput must hence be classified into an output category.

[2] ISL p. 24 [2.13] [?ISL]
[3] ISL p. 26 [2.1.3] [?ISL]
[4] ISL p. 27 [2.1.4] [?ISL]

When output is continuous a modelling problemm will be known as regression, because a model is regressed onto/around the observed points.

### Not alaways clear

It is not always clear whether or not a problem is classification or regression, for example:

Logistic Regression is often used to deal with categorical output using predictors that are either categorical or continuous, for that reason it is considered a classification technique, however logistic regression returns continuous probability values that are then interpreted and could also be used for continuous output, so it could also be considered a regression technique.

# Assessing Model Accuracy

The real trick in data science is deciding which model to use where, ontop of deciding which function best describes the behaviour of the data, the previous considerations of interpretability and capacity for future predictions must also be considered.

## Quality of Fit

in order to assess the performance of a model it is necessary to measure how well the models predictions match the observed data, the most commonly used measure in regression is the *Residual Sum of Squares* (***RSS***), *Mean Square Error* (***MSE***) and *Root Mean Square Error* (***RMSE***):

$$\text{RSS} = \sum_{i=1}^{n} \left[ (y_i - \hat{y}_i)^2 \right] \tag{1.2}$$

$$\text{MSE} = \frac{RSS}{n} = \frac{1}{n} \sum_{i=1}^{n} \left[ (y_i - \hat{y}_i)^2 \right] \tag{1.3}$$

$$\text{RMSE} = \sigma_\varepsilon = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left[ (y_i - \hat{y}_i)^2 \right]} \tag{1.4}$$

Another common Measurement used is the *Relative Squared Error* (***RSE***), this has the advantage that it may be compared between different data sets:

$$\text{RSE} = \frac{\text{RSS}}{\text{Variance}} = \frac{\sum_{i=1}^{n} \left[ (y_i - \hat{y}_i)^2 \right]}{\sum_{i=1}^{n} \left[ (y_i - \hat{y})^2 \right]} \tag{1.5}$$

In effect the (***RMSE***) is the sample standard deviation of the residuals ($\varepsilon$), the standard deviation of a parameter (e.g. $\mu$, $\varepsilon$) is known as *Standard Error* (***S.E.***), so ocassionally (***RMSE***) is known as the *Residual Standard Error* (***RSE***), I don't use that because it's ambiuouswith *Relative Standard Error*.

The (***RMSE***) is an example of a *Loss Function* because it measures the loss between observed data and the model; So for example we may minimize the ***RSS*** on the training data and then use the ***RMSE*** to assess the model performance on the testing data [5]

The model accuracy must be assessed on the testing data because the model has already been optimised for the training data, if the model is a poor choice (say for example because it is overparameterised) it will perform poorly on the testing data and the model may hence be reconsidered.

An alternative to using a test training split is to use Cross Validation.

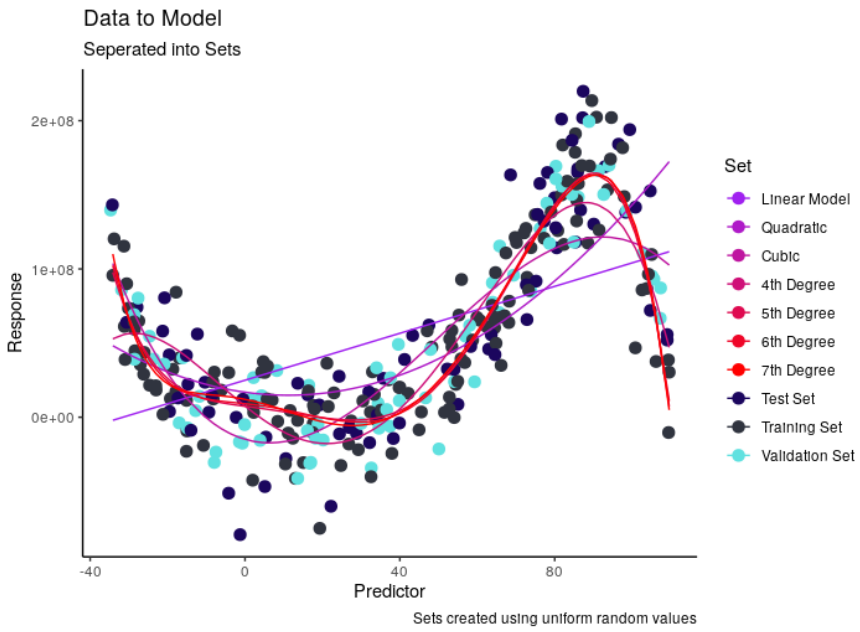



Figure 1.1: Various Models for the observed Data, some of these predict too poorly and some of these are overparameterised

## Model Evaluation

Say for example we had the data and potential Models as in Figure 1.1, if we knew that the data would most certainly follow some polynomial distribution, then it would be appropriate

[5] We minimise the ***RSS*** because it is equaivalent to minimising the ***RMSE*** in the case o<2>f linear regression and an analytic solution via calculus is available (i.e. ***RSS*** simpler to solve than $\sqrt{\boldsymbol{RSS}}/n$), I could just have equaly said 'minimise the ***RMSE***', I merely wanted to draw attention to it.

to consider the $r^2$ value for the *Ordinary Least Squares* linear regression for the models $y \sim x^2$, $y \sim x^3$, $y \sim x^4$ etc and simply choose the most linear.

If however we had no evidence that the distribution would most certainly follow a polynomial distribution yet we desired to fit such a model to the data, it would be necessary to balance overparameterisation with error.

**Test/Validation/Training** one such method is the train/validation/test split, in figure 1.1 the blue data represents the validation data and the darker points are the training data. The models are created on the training data and validated against the blue data that they hadn't yet seen.

The ***RMSE*** is compared to the the degree of the polynomial and the 'saturation' point is chosen as in figure 1.2.

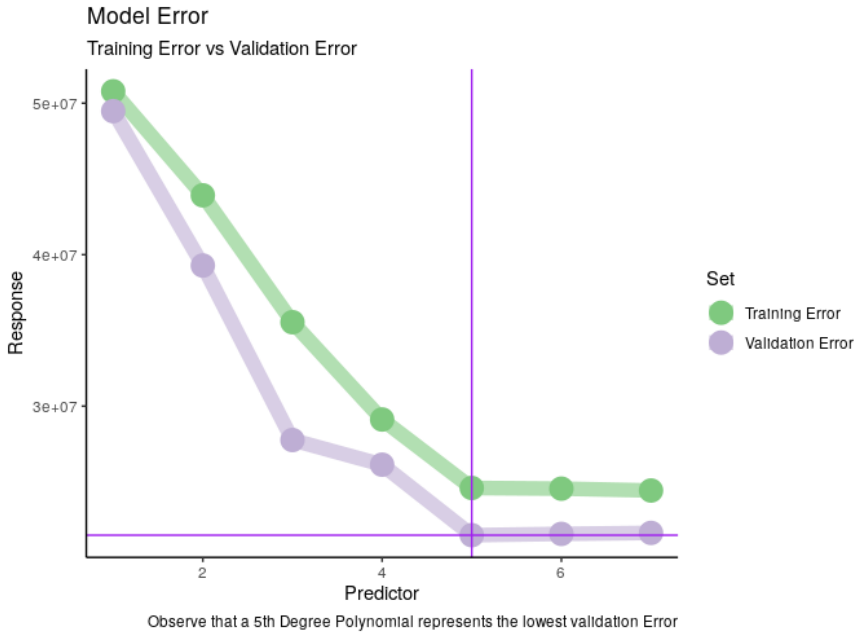



Figure 1.2: Evaluating which model to choose based on the training and validation error in comparison to the degree of the polynomial, this is illustrated far better at p. 36 of ISL [?ISL]

**Cross Validation** If however the data set we had was too small to split the data up, it may be more appropriate to implement Cross Validation, which involves splitting up the data into three groups and then creating and comparing models between the three groups as depicted in figures :

This is compared with the ordinary Test/Train Split as depicted in figure 1.6:

If a model has been overparameterised, we would expect the training error to decrease whilst the testing and validatoin error increase.

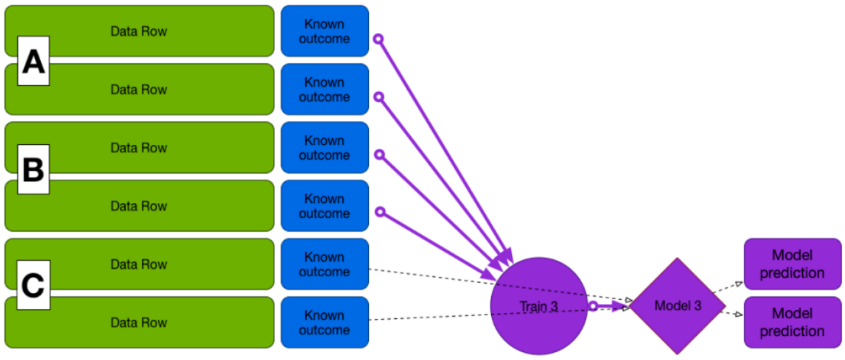



Figure 1.3: First Step in Cross Validation

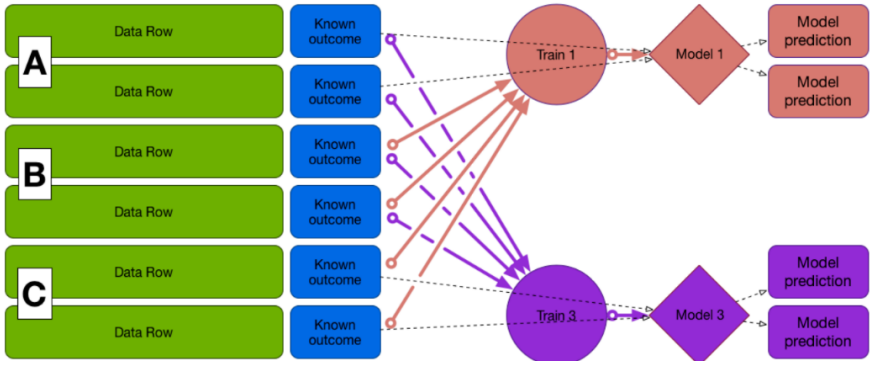



Figure 1.4: Second Step in Cross Validation

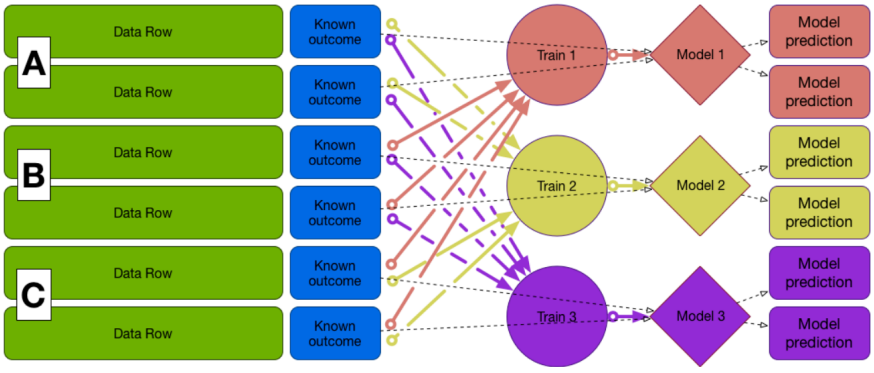



Figure 1.5: Final Step in Cross Validatoin

## Bias and Variance

It can be shown mathematically that the expected *Mean Square Error* is composed of:[6]

[6] ISL p. 34 [2.2.2] [?ISL]

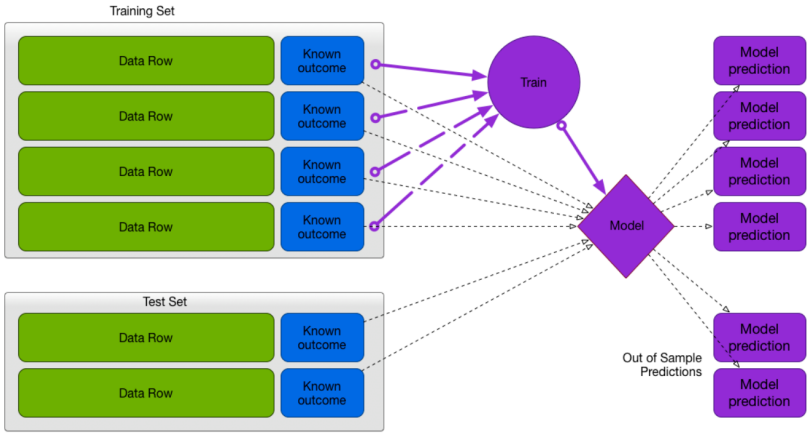



Figure 1.6: Ordinary Test Train Split of Data

$$\mathrm{E}\left(MSE\right) = \mathrm{Var}\left(\hat{y}\right) + \left[\mathrm{Bias}\left(\hat{y}\right)\right]^2 + \mathrm{Var}\left(\varepsilon\right) \tag{1.6}$$

Where:

- Variance of $\hat{y}$ refers to the average squared error we would expect to see if the model was given a different training set
- Bias of $\hat{y}$ refers to the average difference attributed to the choice of model
- $\mathrm{Var}\left(\varepsilon\right) = \boldsymbol{RSS}$ is squared distance from the model attributed to normally distributed random error.

Generally:

- More flexible methods result in less bias but more model variance between sampled data.
  - Connecting every single data point with a $n^{th}$-degree polynomial will have very little bias caused by the model, but there will be a lot of variance between given sets of input.
- less flexible methods will result in more model bias but less variance between sampled data.
  - e.g. drawing a straight line through points will give outputs that a very biased by the model, the error caused between different sets of input will be small.

This is why we have the tendency for testing error to increase as training error decreases when a model is overparameterised.

The trick is finding the right balance between variance and bias.

## Accuracy in Classification

Basically in this context we use the sum of misclassifications, it's well worth reading [2.2.3] of ISL [?ISL] for the demonstration of $k$-*Nearest-Neighbours* and discussion of the *Bayes Classifier*.

# Introduction to R

# Linear Regression

## Simple Linear Regression

The Least Squares regression with the standard errors represents the the range of lines we would get (95% of the time) if we kept taking population samples and fitting an *Ordinary Least Squares* regression.[1]

### Estimating the Coefficients

The coefficients of a linear model are calculated by solving, via calculus, the coefficients that correspond to the minimum value of the ***RSS***.

### Assessing Coefficient Accuracy

The standard error is the standard deviation of a parameter.

Just like the variance of the expected mean value can be determined by:

$$\mathrm{SE}\left(\hat{\mu}\right) = \frac{\sigma^2}{n} \tag{2.1}$$

We can extend this to determine the standard error of the coefficients:

$$\mathrm{SE}\left(\hat{\beta}_0\right)^2 = \sigma^2 \cdot \left[\frac{1}{n} + \frac{\overline{x}^2}{\sum_{i=1}^{n}\left(x_i - \overline{x}\right)^2}\right], \qquad \mathrm{SE}\left(\beta_1^2\right)^2 = \frac{\sigma^2}{\sum_{i=1}^{n}\left[\left(x_i - \overline{x}\right)^2\right]} \tag{2.2}$$

and it also follows that a 95% confidence interval will be of the form[2]:

[1] ISL p. 65 [3.1] [?ISL]

[2] That is to say that there will be a 95% probability that the interval will contain the true populatoin value, where the coefficient is the *Student's* $t$-value corresponding to a distribution with $n-2$ degrees of freedom. This

$$\hat{\beta}_0 \pm 1.96 \cdot \text{SE}\left(\hat{\beta}_0\right), \qquad \hat{\beta}_1 \pm 1.96 \cdot \text{SE}\left(\hat{\beta}_1\right) \tag{2.3}$$

These standard errors can also be used to compute a hypothesis test where the null hypothesis is $\text{H}_0 : \ \beta_1 = 0$, which would imply that there is no relaionship between the observations, in this case the tstatistic would be:

$$t_{\text{d.f.}=n-2} = \frac{\hat{\beta}_1 - 0}{\text{SE}\left(\hat{\beta}_1\right)} \tag{2.4}$$

**Interpreting the $p$-value** The $p$-value is the probability of committing a type one error [3] (i.e. saying 'i.e. saying 'something happened' when in fact nothing happened.)

A small $p$-value indicates that it is unlikely to observe such an association merely by chance.

Bear in mind this is distinctly different fron the probability of the alternative hypothesis being true, this is measured using the power of the sample.

## Assessing Model Accuracy

### RMSE

The ***RMSE*** can be used to assess model accuracy as in (1.4).

### Coefficient of Determination $R^2$

The coefficient of Determination is the proportion of variance in the data that is explained by the model:

$$R^2 = \frac{TSS - RSS}{TSS} \tag{2.5}$$

Where:

ofcourse depends on the assumption that the errors are normally (i.e. Gaussian) distributed; It should be noted that 1.96 is the normal value, and represents the limit value of the $t$-distribution when the degrees of freedom are made arbitrarily large.

[3] Incorrectly rejecting the null hypothesis

$$\textbf{\textit{TSS}} = \sum_{i=1}^{n} \left[ (y_i - \overline{y})^2 \right]$$

$$\textbf{\textit{RSS}} = \sum_{i=1}^{n} \left[ (y_i - \hat{y}_i)^2 \right]$$

In the case of simple linear regression the *pearson correlation Coefficient*: [4]

$$r = \frac{\sum_{i=1}^{n} \left[ (x_i - \overline{x}) \cdot (y_i - \overline{y}) \right]}{\sqrt{\sum_{i=1}^{n} \left[ (x_i - \overline{x})^2 \right] \cdot \sum_{i=1}^{n} \left[ (y_i - \overline{y})^2 \right]}} \tag{2.6}$$

Just so happens to be such that $r^2 = R^2$.

# Multiple Linear Regression

This is also calculated using *Ordinary Least Squares*

To decide on important variables use either backwards or forwards selection [5]

# Linear Regression in R

The model should be generated in ***R*** using the following syntax

```
library(MASS)
lm.fit = lm(medv~lstat, data=Boston)
```

Avoid specifying the vectors `Boston$medv` because it will stop the `predict()` command from working properly later down the track, moreover it is inconsistent with `tidyverse` syntax anyway.

## Non-Linear

If you are going to specify a non-linear model, make sure to wrap the coefficients in `I()` because otherwise R will interpret things like `*` and `^` as signals to the model and not as mathematical options between the sets of data.

You can also use `poly()` in order to specify whether or not to return *Orthogonal* or *raw* polynomial coefficients, the orthogonal coefficients lose there direct meaning to the set of data,

[4] p. 70 eq 3.18 of ISL [?ISL]
[5] p. 78 of ISL [?ISL]

however, they are no longer correlated with each other and the $p$-values returned are more meaningful.

# (Wk 4) Classification; Logistic Regression

*Topic 4* / *Chapters 4 ISL* [?ISL]

## Overview of Classification

Linear regression assumes that an ouput value is a continuous variable, often however an output variable is a categorical variable (equivalently a factor, discrete value, qualitative variable) and a model that predicts a categorical variable is known as a classifier.

There are three main techniques used for classificaiton, *Logistic Regression*, *Linear Discriminant Analysis* and $k$-nearest neighbours.

The three most commonly used mo

## Why not Linear Regression

Basically because it has the wrong shape and the probabilities may be greater or less than 1/0

## Logistic Regression

### The Logistic Model

Basically all we do is assume that the $\log(\text{Odds})$, often referred to as the *logit* are linear:

$$\log\left(\frac{\Pr(X)}{1-\Pr(X)}\right) = \beta_0 + \beta_1 \cdot X \tag{3.1}$$

#### Assumptions of Logistic Regression

## Logistic Family; Binomial vs Bernoulli

The Bernoulli and Binomial Distributions form the basis for logistic regression [1]

**Binomial Distribution** There are three characteristics of a binomial experiment [2]:

1. There are a fixed number of $n$ trials
2. There ae only two possible outcomes (wherein $p$ is the probability of success, and $q$ the probability of failure)
3. The trials are independent and repeated with identical conditions.

The Bernoulli Distribution is the binomial distribution for only a single trial.

The histogram of the number of successes from a binomial distribution may be made arbitrarily close to a normal distribution so far as the sample size is made sufficiently large, after 5 samples it is considered normal [3].

It is computationally easier than the `logit` link

## Link Functions; 'logit' vs 'probit'

Some fields prefer the `probit` function by convention, for example toxicology.

`probit` is short for **Pr**obability un**it**, it was first published by chester bliss in 1934 in the field of toxicology for modelling dose response curves.

The `probit` function is Computationally less resource intensive than the `logit` function.

The `logit` curve is given by a linear function:

$$\log\left(\frac{\Pr(X)}{1-\Pr(X)}\right) = \beta_0 + \beta_1 \cdot X \qquad \text{(3.1 revisited)}$$

The probit assumes a binomial distriution and is linked by the equation:

$$\Phi^{-1}(p) = \beta_0 + \beta_1 x + \varepsilon \qquad (3.2)$$

This is based on a cumulative normal distribution:

[1] Refer to DataCamp[?dcampGLMLregV1]
[2] p. 253, [4.3] of Rice Intro Stats [?ricestat]
[3] Real-Statistics.com

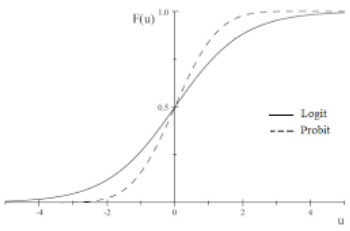



Figure 3.1:

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \cdot \int_{-\infty}^{z} e^{-\frac{1}{2}z^2} \, \mathrm{d}\, z \tag{3.3}$$

Which takes the input $z$ and returns the probability of observing a value less than or equal to $z$.

By looking at the shape it can be seen that the `logit` might be better at modelling outliers.

Sometimes it is helpful to simulate data for analysis, to do this for probit analysis:

1. Convert a $z$-score to a probability by using `pnorm`
    - If you wanted so simulate `logit` you would use `dlogis` here instead to get the corresponding probability for `logit`
2. Use the probability with a binomial distribution to generate 0 and 1 outputs by using `rbinom`

**How to Choose** Either model works well, `logit` has thicker tails so it is slightly better at predicting outliers however either is tenable.

## Assumptions of Logistic Regression

The following assumptions are made when using logistic regression, these limitations also apply to *Poisson* and other *GLM* models as well:

- Simpson's Paradox
    - occurs when an important parameter is missing
    - Inclusion of the predictor will change the outcome
- the predictor variables are linear and monotonic
- predictors variables are independent
- Overdispersion
    - one possible cause of this could be heteroskedastic data

## Estimating the Regression Coefficients

## Making Predictions

# Linear Discriminant Analysis

# KNN

# Quadratic Discriminant Analysis

a

# (Wk 5) Model Selection

*Topic 4* / *Chapters 4 ISL* [?ISL]

## Cross Validation

### Bias and Variance

#### Bias

Bias is the variance in the output data that is attributable to model simplification [1], so for example if we had a model with very high bias (e.g. an inflexible linear model on non-linear data), now matter how much training data was given, the model would still have substantial errors.

High error implies high bias.[2]

#### Variance

"Variance refers to the amount by which the output ata would change if we estimated it using a diferent training set" [3], so for example a linear model fitted to non-linear data would be such that the variance in the output data would be the same regardless of the training data used, so a linear model is an example of a non-flexible model that has low variance.

### Test Validation Training Split

Simply splitting data into a test and validation set as a couple problems:

[1] ISL p. 35 Ch. 2.2 [?ISL]
[2] I couldn't actually here her...
[3] ISL p. 34 Ch. 2.2 [?ISL]

1. The model will be trained on less data, this means the model will return an ***RMSE*** [4] value that will vary greatly depending on how the data is split.

2. The model will also perform worse owin to the smaller set of training data, meaning that the predicted model performance will underestimated (i.e. the model error will be overestimated).

## Leave One Out

This will fit the model to all but one data point and calculate the ***RMSE*** on that single data point, then that 'left-out' data point will be put back into the training data and another selected, the model will be refit and the ***RMSE*** again recalculated on the 'left-out' data point.

This will be repeated until $n$ ***RMSE*** values have been calculated, then those values will be averaged and this will represent the loss function of the model.

This approach has advantages over the training/validation split:

1. The predictoin of the ***RMSE*** is less biased because the model has considered all the data points

2. The training error will no longer be overestimated.

Moreover the error statistic will always be the same value for any given data and model, which is helpful (whereas training/validation split will vary and so will $k$-fold validation).

### Computational Resources

This approach means that a model has to be fit $n$ times, this will be heavy to implement, fortunately there is actually an analytic solution to the ***RMSE*** value calculated by this method that applies to polynomial models, however there is no analytic solution for other models.

This means that if a model is very complex to fit, and/or the data set is very large, this could be very resource intensive to implement .

## $k$-fold Cross Validaiton

Cross validation is very similar to *Leave One Out CV*, basically the data is seperated into $k$ groups [5] and one of those groups becomes the validation group.

[4] as in the standard deviation of the error, this is our typical loss function

[5] Emperically 5 or 10 groups performs the best

The data is fit on the remaining data and the ***RMSE*** is calculated on the validation group.

Just like the *Leave One Out CV* method this is repeated $k$ times and then the $k$ ***RMSE*** values are averaged.

**Advantagages** This has the advantage of being far less resource intensive to implement that *Leave One Out CV* but also it tends to perform better, The validatoin split estimates the testing error with a lot of variance, *Leave One Out CV* estimates the testing error with to omuch bias, $k$-fold (using 5/10) fold tends to be the best compromise.

**How many Folds** Generally 10-fold cross validatoin is more common.

### Evaluation Performance

A Test Training Split tends to be vary Biased when predicting the testing error and *Leave One Out CV* tends not to be, however as a trade of the *Leave One Out CV* method suffers from high variance, in practice 5/10-fold CV tends to be the best method to use in order to evaulate model performance. [6]

## Bootstrapping

If we want to know the expected uncertainty with a model, it can be difficult to determine.

> e.g. if the assumptions of linear regression are satisfied then we may easily know the standard error of $\beta_0$ and $\beta_1$, however if we had a *KNN* model what on earth would the standard error of $\hat{y}$ be?

We might like to simulate the data and then test the model performance, but that won't work because if we knew how to simulate the data we would know how the population behaves and we wouldn't have to worry about choosing a model, we could simply evaluate the standard deviation of the population.

However if we realise that simulating the data is essentially sampling data from the population we can make the connection that when we don't have a population statistic we infer it from the population.

Hence, instead of sampling the population, sample the training data (yeah, you're resampling the sample).

So say we are concerned with the standard error of our predicted value of $\alpha$ from our trianing set of size $n$:

1. Create a new sample of size $n$ wherein each observation has an equal probability of being equivilant to the any observations in the original data set

[6] ISL p. 183 Ch. 5.1.4 [?ISL]

(a) This is not a permutation, reptetition is allowed, in theory any sample could all contain the same value, but in practice the probability of this occuring is too low to happen.

2. Create this new sample $B$ times.
3. Take the value of $\hat{\alpha}$ from each of the new samples.
4. the standard deviation of $\hat{\alpha}$ from those $B$ samples is approximately the standard deviation of the predicted $\hat{\alpha}$ from the population data.

### Mean, Median and Mode

- The median is the middle value of a dataset
- the mode is the most frequent bin value
- arithmetic mean is the value that may be added $n$ times to give the final value.

# Mathematical Modelling

Often times the degree of a polynomial is used to represent the flexibility of the data, if we actually knew the data was polynomial, we would be better off linearising the data first and testing for linearity. however we do not know that the data will most certainly follow a polynomial distribution.

# Boostrapping and Cross Validatoin

- Use cross validation to predict the expected variation in the output data
  - Recall that the expected variation in the output data is standard error of $\hat{y}$ which is S.E. $(\hat{y}) = \sigma_{\hat{y}} =$ ***RMSE***
- use Bootstrapping to predict the standard error of any other model parameter

# (Wk 6) Tree Based Methods

*Topic 6* / *Chapters 8.1 ISL* [?ISL]

## Regression Trees [8.1.1]

## Classification Trees [8.1.2]

- Use Misclassification rate not deviance

## Trees vs Linear Models [8.1.3]

- Trees are better for really complicated relationships
- they may be better for situations that reflect human behaviour because trees are a closer analogy to human thought than regression

## Advantages and Disadvantages of Trees [8.1.4]

- simple
- interpretable
- Easy to Plut
- Don't have to worry about dummy Variables

But generally less accurate than other models. This is better for trying to interpret the relationship between data than forecasting results.

# Revision

- We have continuous and discrete response data which determines whether or not we would use regression or classification.
- Structured vs Unstructured data, I forget?

## Regression

In weeks 1,3,4 we focused on linear, multiple and logistic regression.

$$y = \alpha + \beta_1 X_1 + \beta_2 X_2 \tag{6.1}$$
$$\hat{y} = f(X) \tag{6.2}$$

in simple linear regression use one predictor in order to predict output variables:

$$\text{SLR:} \quad \hat{y} = \hat{\alpha} + \hat{\beta} x \tag{6.3}$$
$$\text{MLR:} \quad \hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_2 x + \cdots \hat{\beta}_p x_p \tag{6.4}$$

### Simple Linear Regression

1. Identify Response
2. Use a scatter plot to visualise the most linear
3. also use the correlation coefficient
4. then model evauluation
   (a) hypotheses testing using the $p$-values
   (b) Check the significance of the parameter estimates
   (c) overall accuracy of the model
      i. $R^2$-value, adjusted $R^2$-value, *Residual Standard Error*.

   Assumptions

- $\varepsilon\ N(0,\sigma), \quad \exists\sigma \in \mathbb{R}$
- independent residuals
- Predictors are measured without error.

(d) In order to evaluate this you would use `plot(lm(Y~X))`

5. next you would be interested in making predictions from the model

## Multiple Linear Regression

1. Check the significance between variables using a `cor()` matrix.
2. create a scatter plot matrix
   (a) look for the significance of the relationship between the response and the feature
      i. Is it Linear?
      ii. is it a strong relationship?
3. same as simple linear regression, don't forget to write out the equation
4. Polynomial Regression:
   (a) In order to determine if a term is polynomial, analyse the scatter plots.
   (b) In order to determine which degree, the simplest way is to increase the degree until the LAST degree is significant (not the anyother, because it's an even/odd shape thing.) so all the terms will be kept until the FINAL term ($x^n$) is not significant, keep all preceeding terms).
   (c) If it violates the assumptions of the linear regression we can use a transformation of $\log_e(y), \frac{1}{y}, e^y, \log_e\left(\frac{p}{1-p}\right)$

## Logistic Regression

1. check the relationship between the resoponse and the variables.
2. instead of using a scatterplot it makes more sense to use a boxplot.
   (a) Response on the bottom
   (b) multiple plots corresponding to the different features plotted along the $y$-axis.
3. RSS is a special case of maximum likelyhood function
4. Write the equation:

$$\hat{P}(X) = \frac{e^{\hat{\alpha}+\hat{\beta}x}}{1+e^{\hat{\alpha}+\hat{\beta}x}}$$

## Resampling

Here we are concerned with avoiding overfitting the model, we have a few methods discussed below, generally the best method is 10-Fold Cross-Validation.

### Test and Training

Usually use more data for the training data, otherwise the error will be overestimated, usually use 70%/30%.

Use the training data to fit the model and then test the data, however because the sample is random we will get a different number each time.

### Leave-one-out Cross-validation (LOOCV)

This will fit the data to $n-1$ points and then test it on the left out point, this will never change but it has two problems:

1. Underestimates the error
2. Is resource intensive

### 10/5-Fold-Cross-Validation

This is the best way to go, the best compromise on resources and under/over estimation.

# Trees

1. Fit the tree
2. then use cross validation to determine the best number of nudes/leaves, this is called pruning
   (a) to acheive this we use `cv.tree()`
3. A classification tree is used by just dividing the data up into averages and boxes.
4. Performance Measures:
   (a) Classification tree; Misclassification Matrix
   (b) Regression Tree; Mean Square Error??

# Support Vector Machines

Support vector machines is a classification problem.

We are trying to fit a hyperplane in $p$-dimensional space, we use the `tune()` to perform cross validation to decide on the best parameters for the hyperplane.

In order to decide on the best kernel, fit all three kernels and then CV will automatically choose the best parameters, we just have to choose the kernels, use the loss function to decide on which model to choose.

does the cross validation fit the parameters to the hyperplane, no that would be minimising the RSS, there is a tuning parameter in the loss function/*RSS* and the cross validation chooses the best tuning parameter.

SVM is only a classification technique, misclassification rate is the loss function, the linear hyperplane is fit using the RSS usually (even if the feature space is transformed).

# PCA

We measure the distance from the line, not, parallel.

## Scree Plot

This is a model of $Y$ $X$ such that Variance No. of PC's

Biplot is a scatter plot of the variables along the principal components to see what the clustering might imply.

Scaling of the variales; always scale the variables, or standardise with $\mu = 0$ and $\sigma = 1$

### Deciding on how many PC's

Use `summary(pc.mod)` and then read where the cut off on variance explanation is significantly improved. and/or use the scree plot.

which variables are explained by the first principle component, which variables are explained by the second principle component, this can be interpreted by looking at the PC loading vectors:

$$\begin{pmatrix} \text{Sepal Length} & 0.36 & -0.65 & 0.58 & 0.31 \\ \text{Sepal Width} & -0.08 & 0.73 & -0.579 & -0.319 \\ \text{PL} & 0.36 & -0.65 & 0.58 & 0.31 \\ \text{PW} & 0.36 & -0.65 & 0.58 & 0.31 \end{pmatrix}$$

this would give us:

$$\text{PC1} = 0.361 \times SL + 0.36 \times \text{SW} + 0.58 \times \text{PL} + 0.35 \times \text{PW} \cdots$$

# $k$-means Clustering

In $k$-means clustering we usually use *Euclidean*-distance, there are however other measurements of distance (Manhattan, city-block etc.).

for hierarchical clustering we measure the distances and then use a link-funtion to join the observations together (single, complete, average) then we draw a dendogram.

$$p\left(X\right) = \frac{e^{\beta_0+\beta_1 X_1+\cdots\beta_p Xp}}{1+e^{\beta_0+\beta_1 X_1+\cdots\beta_p Xp}}$$