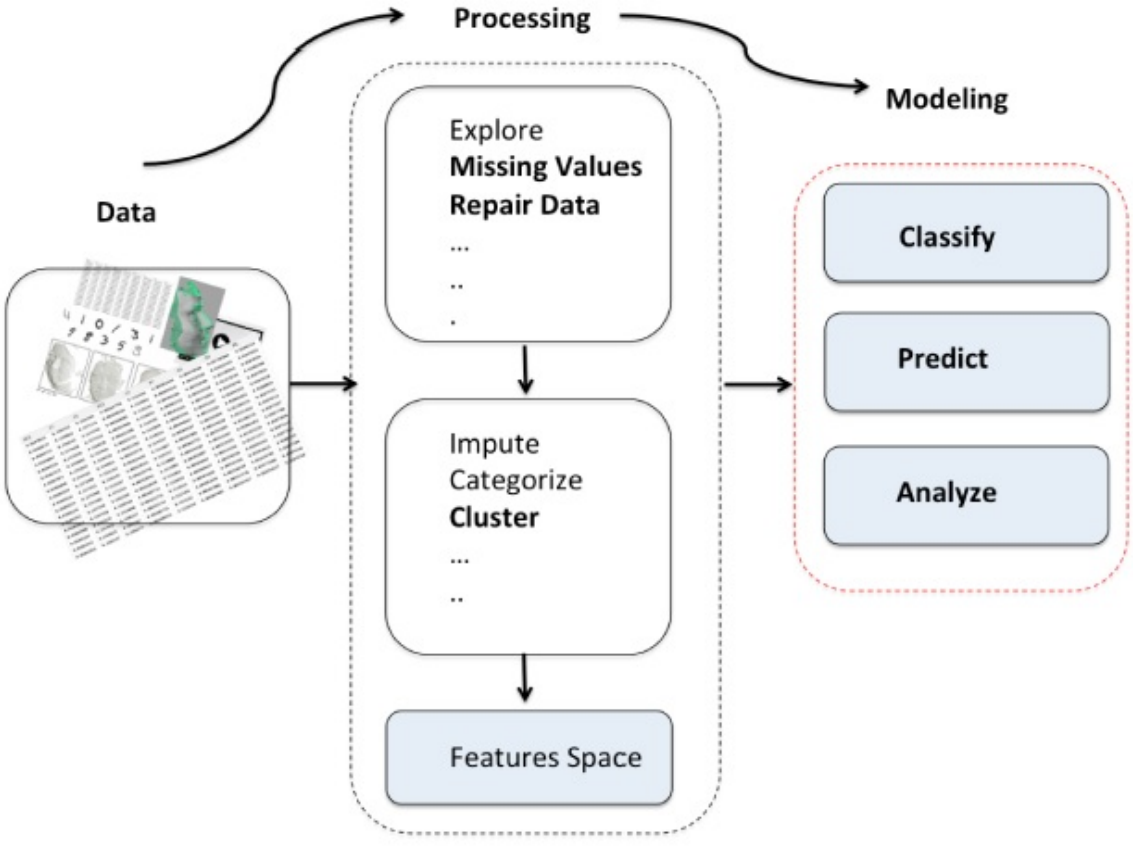
Introduction to R Workshop (Session 3)

Dr Carlos Moreno-García

School of Computing
Robert Gordon University

Machine Learning - Overview



Machine Learning - Terminology

- Feature Space
- Class or Target
- Supervised Machine Learning
- Unsupervised Machine Learning
- Training Set
- Testing Set

Machine Learning - Formal Definitions

A dataset A with m instances $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m$, where each instance $\mathbf{x}_i$ is defined by an n features as $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{in})$.
In a typical supervised machine learning scenario, these instances are often labelled or categorised.

$$A = \begin{bmatrix} x_{11} & x_{12} & ..., & x_{1n} \\ ... & x_{22} & ..., & ... \\ ... & ... & ... & ... \\ x_{m1} & ... & ..., & x_{mn} \end{bmatrix}, Y = \begin{bmatrix} y_1 \\ .. \\ .. \\ y_m \end{bmatrix} \quad (1)$$

Learn a function $h(x)$ that maps an instance $\mathbf{x}_i \in A$ to a class $\mathbf{y}_j \in Y$.
Notice that if Y is a set of discrete values then we call this a **classification** problem, otherwise it is called a **regression** problem.

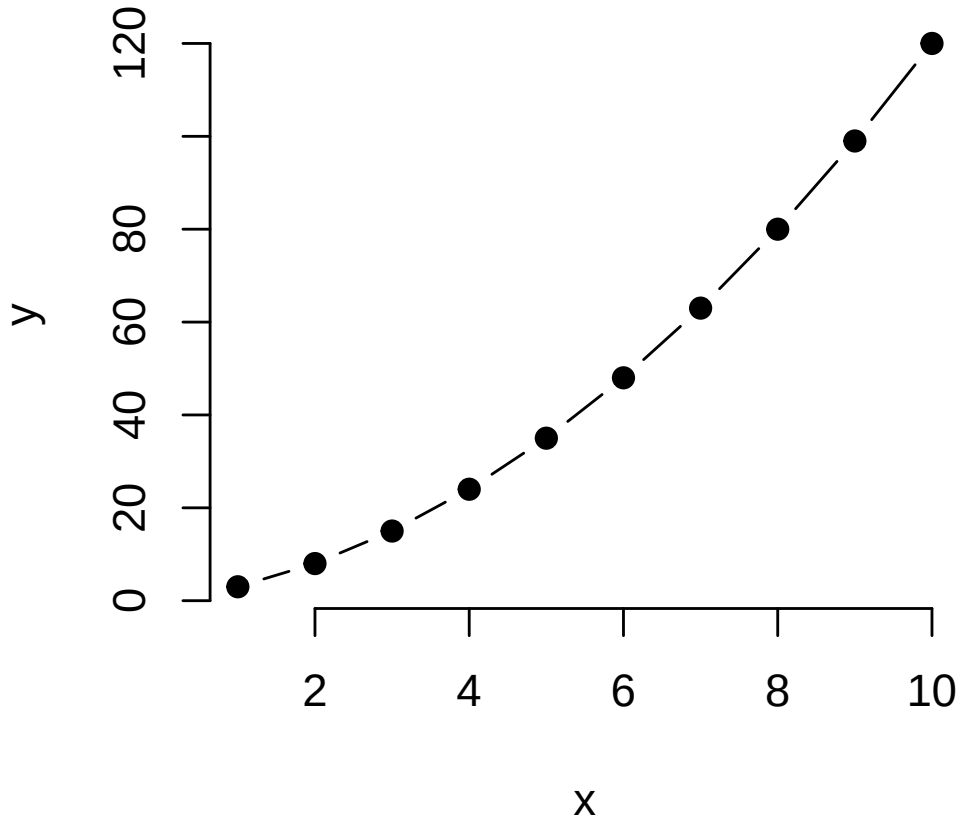
Outline

Regressions

A regression is possibly the simplest form of machine learning. It is based on establishing a function based on a set of points, where x is the feature and y is the target.

Regression - Example 1

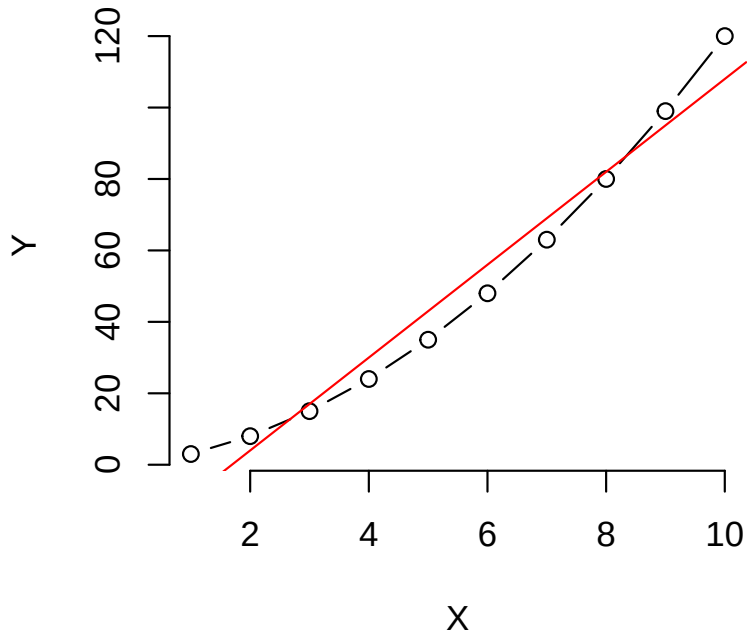
No	X	Y
1	1	3.00
2	2	8.00
3	3	15.00
4	4	24.00
...	...	...
18	18	360.00
19	19	399.00
20	20	440.00



Regression - Example 1

Linear Regression $h(x_0) = \theta_0 + \theta_1 x$

```
x <- seq(1:10)
y <- 2*x +(x*x)
df <- data.frame(X=x, Y=y)
plot(df$X,df$Y,type='b',
     pch=21,frame=FALSE)
fit <- lm( Y ~ X,data=df)
abline(fit,col='red')
```



Regression - Example 1

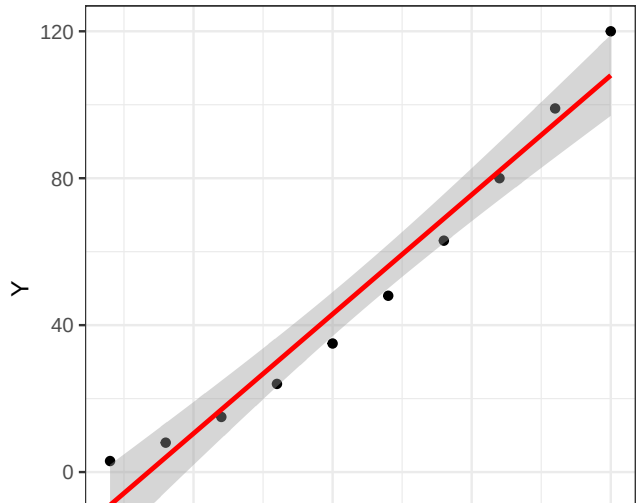
ROBERT GORDON UNIVERSITY ABERDEEN

Regressions with ggplot

```
require(ggplot2)
x <- seq(1:10)
y <- 2*x +(x*x)
df <- data.frame(X=x, Y=y)
```

```
ggplot(df, aes(x = X, y = Y)) +
  geom_point() +
  stat_smooth(method = "lm",
              col = "red") +
    theme_bw()

## `geom_smooth()` using
formula 'y ~ x'
```



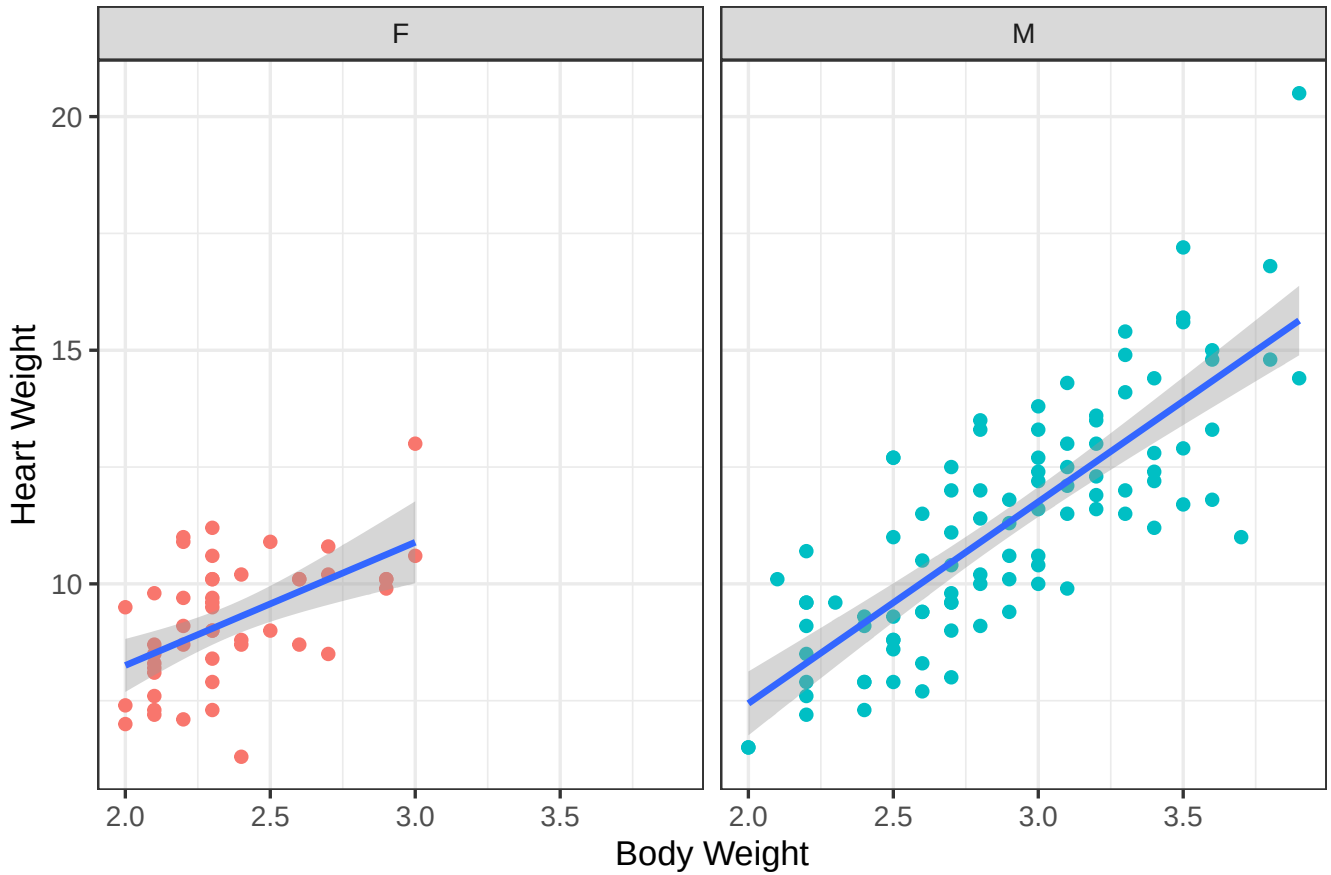
Regression - Example 2

Consider the *cats* dataset.

```
library(MASS)
data(cats)
require(ggplot2)
p <-  ggplot(cats, aes(x = Bwt, y = Hwt))
p <- p + labs(x="Body Weight",y="Heart Weight")
p <- p + geom_point(aes(col=Sex))
p <- p + stat_smooth(method = "lm")
p <- p + facet_wrap(~Sex)
p <- p + theme_bw()
p <- p + theme(legend.title = element_blank())
p <- p + theme(legend.position='none')
p
```

Regression - Example 2

```
## `geom_smooth()` using formula 'y ~ x'
```



K-means

K-means clustering is a method of classifying/grouping items into k groups, where k is the number of clusters). The grouping is done by minimizing the sum of squared distances (i.e. Euclidean distance). This method (in its most basic form) does NOT take the target into consideration.

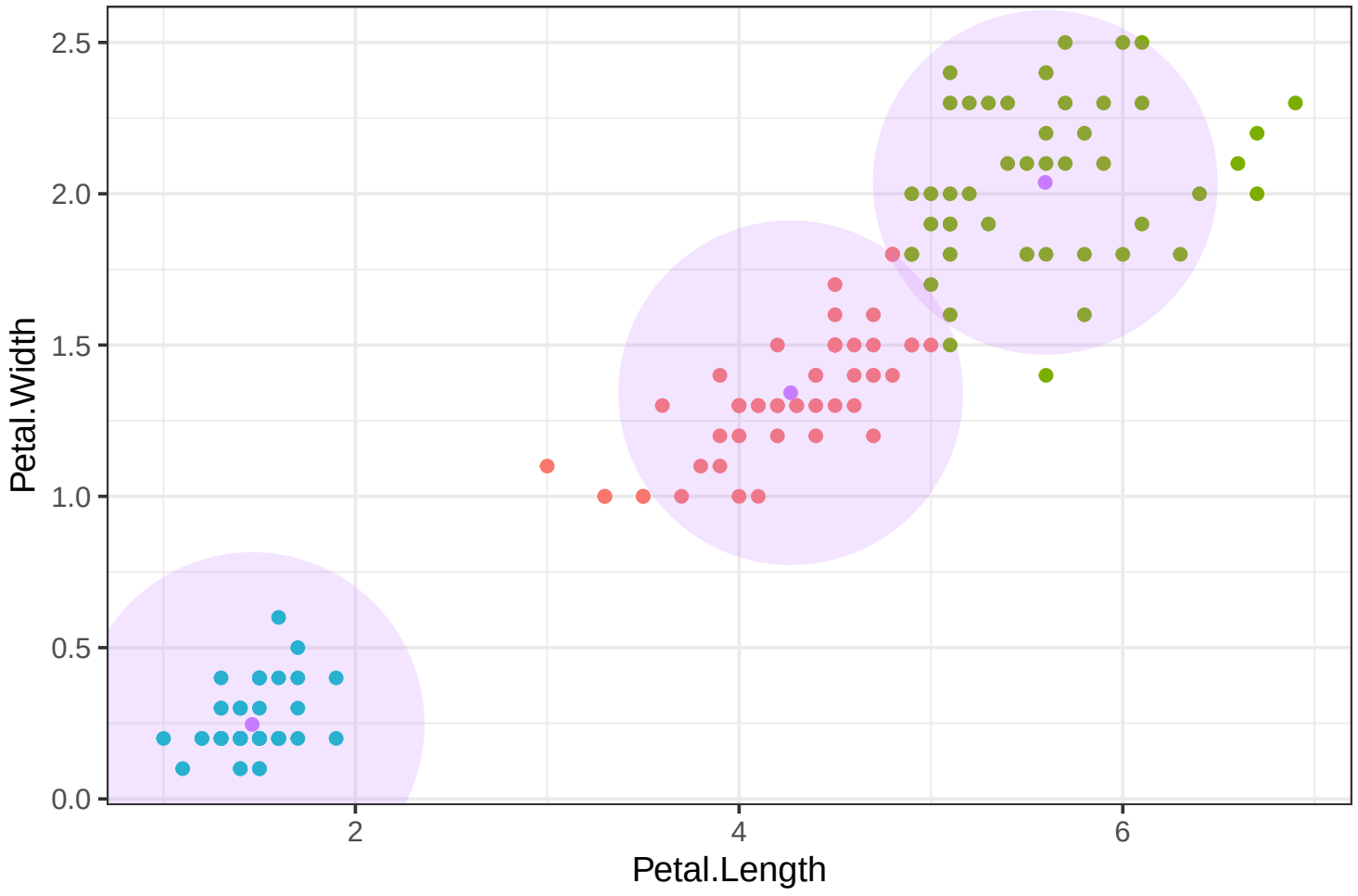
Clustering - Example

Using the Petal.width and Petal.length features of the *iris* dataset, We will use the *ggplot2* package to apply k-means to cluster and visualise the data.

Clustering - Example


```
df=iris
m=as.matrix(cbind(df$Sepal.Length,
        df$Petal.Width),ncol=2)
cl=(kmeans(m,3))
df$cluster=factor(cl$cluster)
centers=as.data.frame(cl$centers)
p <- ggplot(data=df, aes(x=Petal.Length,
       y=Petal.Width,color=cluster),size=.2,alpha=.4)
     + geom_point() + geom_point(data=centers,
                aes(x=V1,y=V2, color='Center')) +
     geom_point(data=centers,
                aes(x=V1,y=V2, color='Center'),
                size=50, alpha=.2) +theme_bw()
p <- p + theme(legend.title = element_blank())
p <- p + theme(legend.position='none')
p
```

Clustering - Example



Final advise

- Split your data into test, train (and validation).
- Select the right feature set.
- There is a handful of machine learning techniques!