Introduction to

Machine Learning

Overview of common algorithms

Content

- **Supervised Learning**
  - Regression
    - Linear (Simple and Multiple)
    - Polynomial
  - Classification
    - Logistic Regression (Binomial and Multinomial)
    - kNN
    - SVM
    - Decision Trees
- **Regularization**
  - Ridge Regression
  - Lasso Regression
  - Elastic-Net Regression
- **Ensemble Methods**
  - Bagging
  - RandomForest
- **Adaptive Boosting**
- **Unsupervised Learning**
  - Clustering

Types of data

Quantitative (Numeric)		Qualitative (Categorical)		
Discrete	**Continuous**	**Binary**	**Nominal**	**Ordinal**
5	5.3	0	Category A	Poor
23	5.9	1	Category B	Good
66	6	1	Category C	Best
41	6.33	0	Category D	Excellent

Can take distinct values

Can take up any value

Takes up only two categories

Categories with no relation

Categories with relation

Terminologies

1. **Predictor variable** – or Explanatory variable or Independent variable (generally referred by X)
2. **Response variable** – or Dependent variable (generally referred by y)
3. **Nominal variable** – or Categorical variable (e.g. Person Names, Gender, etc.)
4. **Population** – Total number of observations available. (usually not considered for analysis)
5. **Sample** – A small part of population that is considered for analysis.
6. **Correlation** – defines the dependency of one variable on another. It (Pearson's correlation) exists between [-1, 1].
   a) 1 -> if both the variables are directly proportional to each other
   b) -1 -> if both are inversely proportional
   c) 0 -> if there occurs no relationship between the variables.
7. **Confidence Interval (CI)** – An $x\%$ of CI estimates that there is $x\%$ surety that a variable will occur in an interval defined by CI i.e. $[Lower\ bound, Upper\ bound]$

Terminologies (Contd.)

8. Standard Error of the estimate (SEE)/Residual Sum of the Squares (RSS) – defines the square root of the average of the squared error of the prediction.

Actual value	Predicted Value	Error (Residuals)	Squared Error
2	2.5	0.5	0.25
5	6	1	1
8	7	-1	1

$$RSS = \frac{0.25 + 1 + 1}{3} = 0.75$$

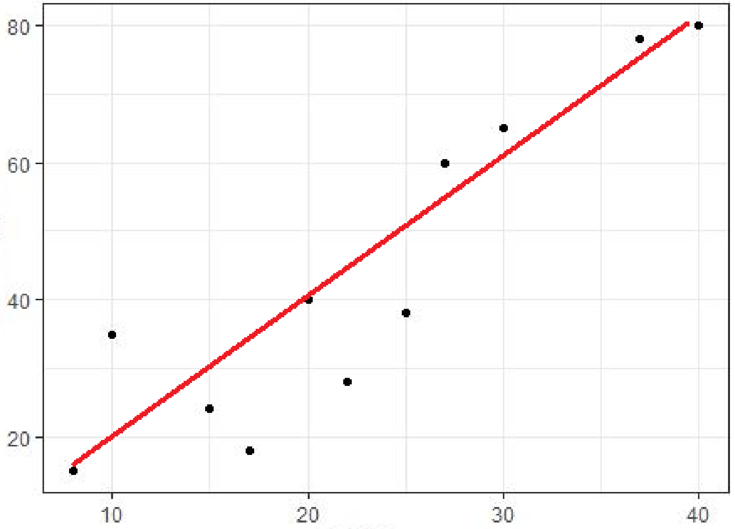
9. Regression Coefficient – defines the mean change in the response variable with a unit change in the predictor variable, given other predictors are kept constant. For instance, $y = 21a + 56b + c$. Here, if variable b is kept constant, then with every unit rise in variable a, y increases by 21 on average.

10. Regression Intercept – defines the mean value if all the predictors are removed (values equated to zero)

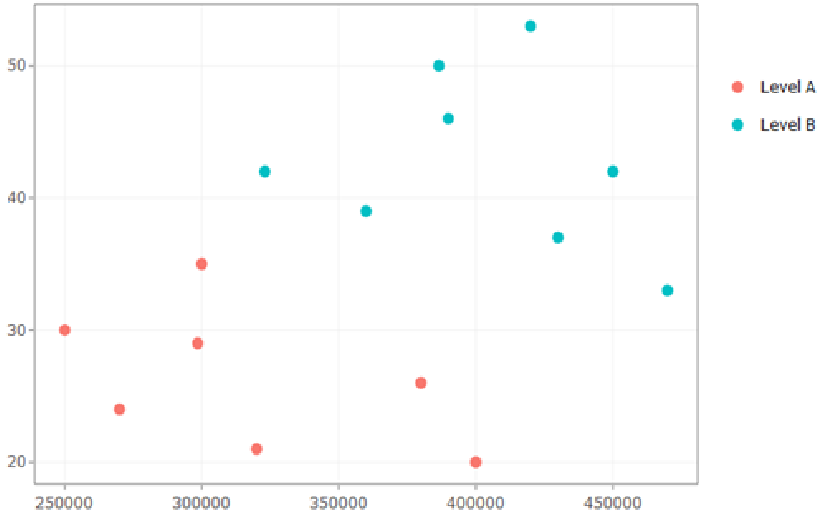
Supervised Machine learning Types

Variables	Regression	Classification
Response (Dependent variable)	Quantitative	Qualitative
Predictor (Independent variable)	No restriction	No restriction

Regression



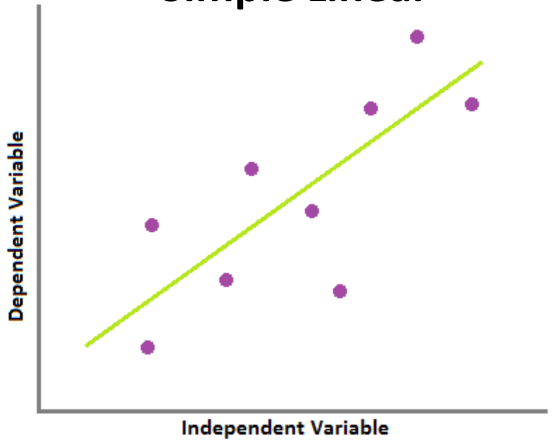
Classification



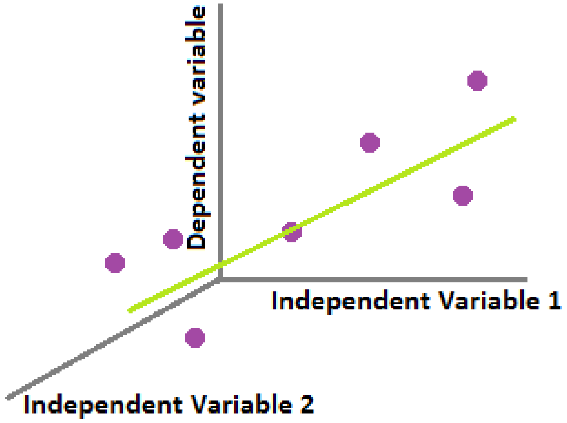
Regression

- A regression technique estimates new values based on the model built on the data fetched from past values (a case of supervised technique).
- Broadly, regression can be divided into three type –

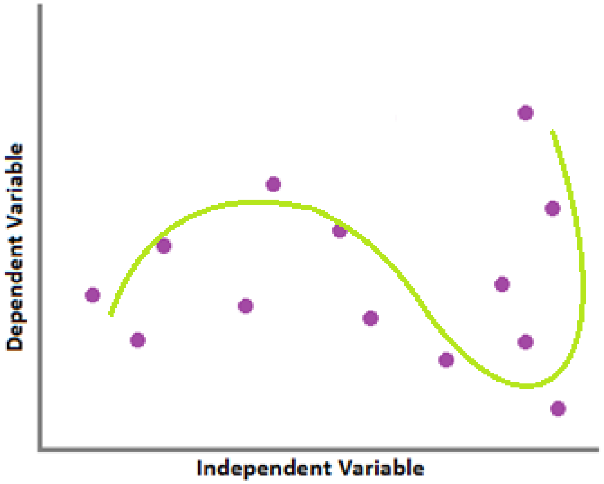
Simple Linear



Multiple Linear



Polynomial



Simple Linear Regression

Consider we have been given with the actual values of a variable (say) X, then a linear regression model response variable $\hat{y}$ is given as:

$$\hat{y} = \beta_0 + \beta_1 X$$

The values of β can be calculated using *Gradient Descent* or *Coordinated Descent* methods.

The best model is the one which has **least RSS**. Other parameters like Adjusted R^2, AIC, BIC, etc. can also be used.

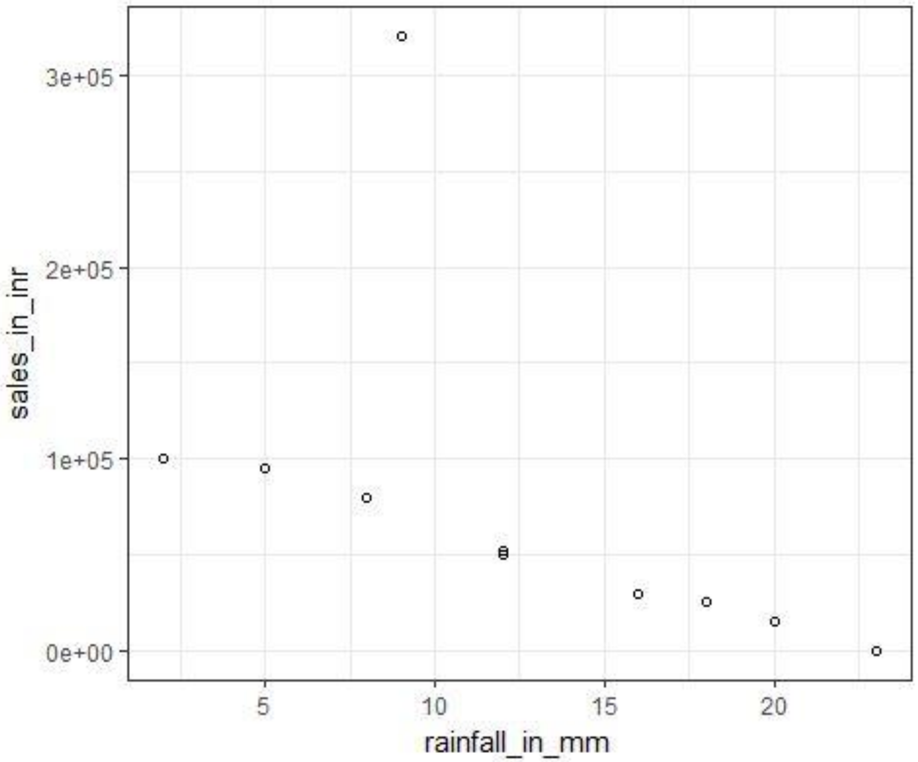
Simple Linear Regression – Illustration

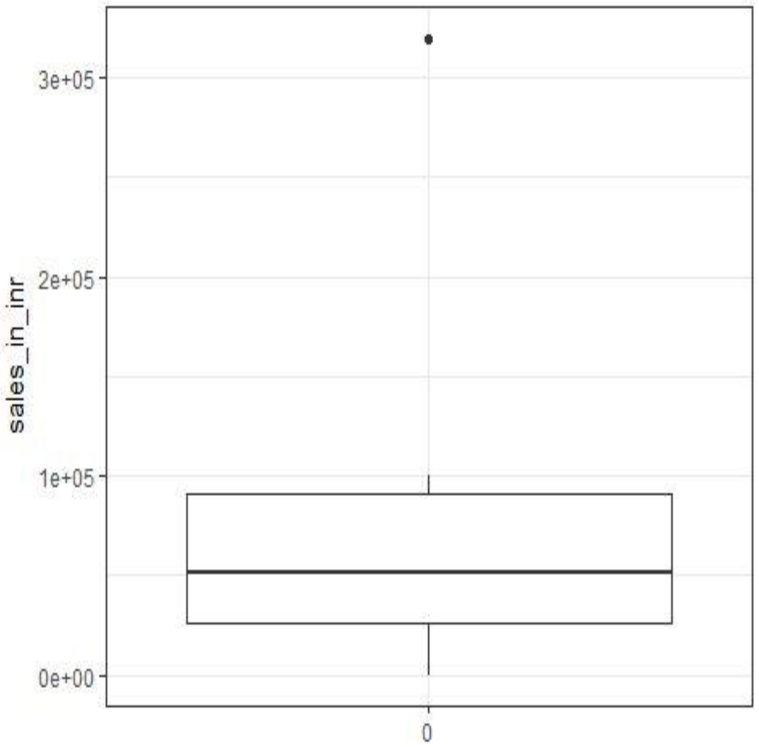
Consider sample of a dataset shown here.

Our task is to determine sales based on rainfall.

	rainfall_in_mm	sales_in_inr
1	2	100000
2	5	95000
3	8	80000
4	9	320000
5	12	50000
6	12	52000

Visualizing the data





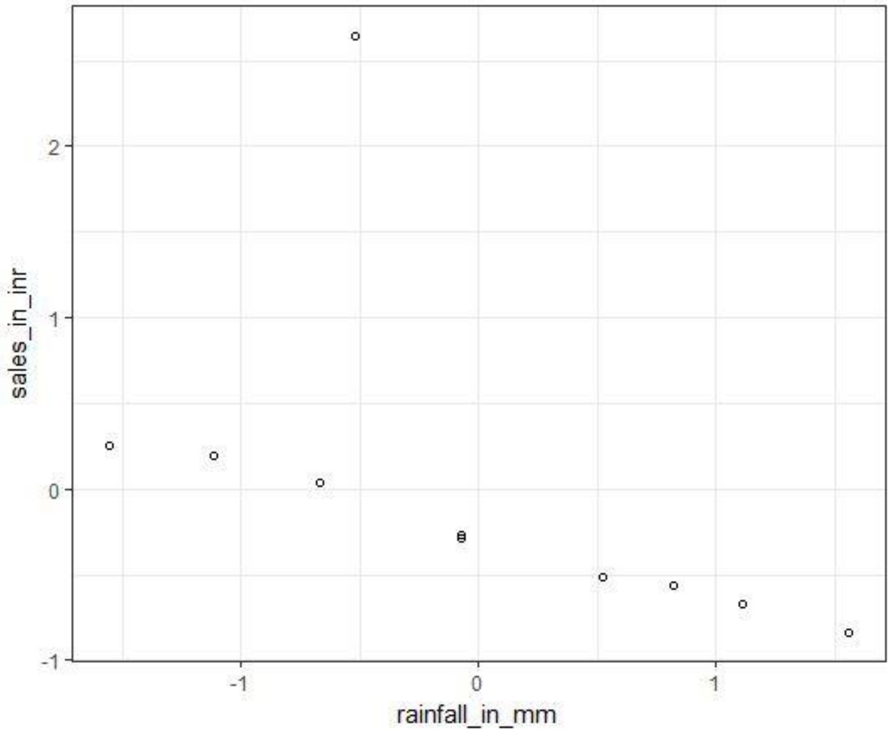
Insights:

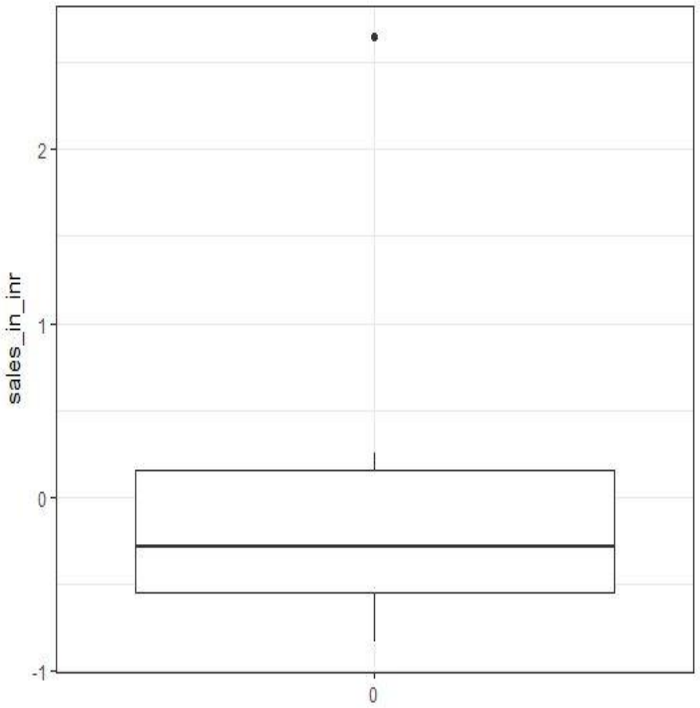
Linear pattern (negative)

Unequal scales

Data with an Outlier

Visualizing the data – after normalization





Insights: Structure remains the same. But scales are normalized.

Verifying relationship using correlation matrix

Correlation as discussed previously under terminology section provides relationship among the variables.

	rainfall_in_mm	sales_in_inr
rainfall_in_mm	1	-0.52742
sales_in_inr	-0.52742	1

Insights: There is a negative relationship between both the variables. However, the relationship is neither too strong (doesn’t reach -1) nor too weak (doesn’t reach 0)

Fitting a linear regression model

A simple linear regression model can be fit on the given dataset and following results are expected –

Coefficients:

(Intercept)	rainfall_in_mm
7.599e-17	-5.274e-01

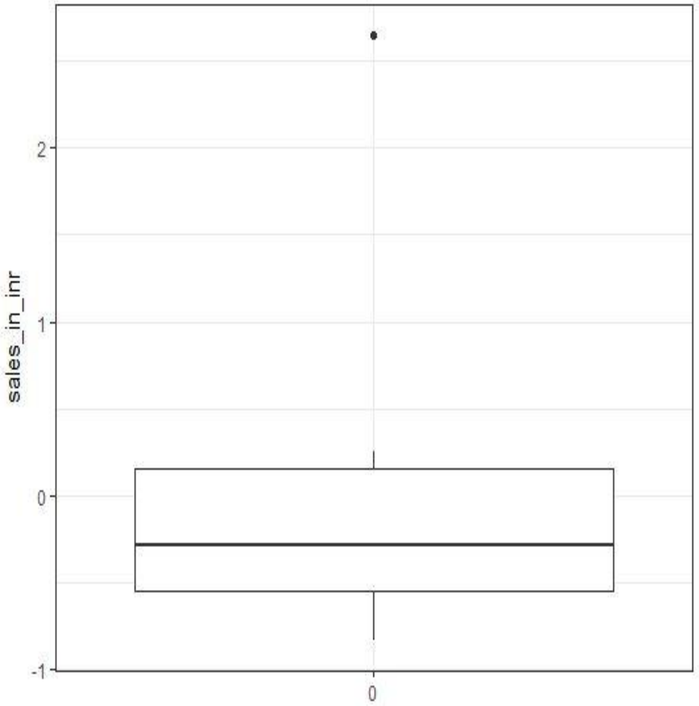
RSS: 0.6496

Insights: We're able to obtain a very small RSS. But we observed that our data has an outlier. Let us try to remove it and rebuild the model.

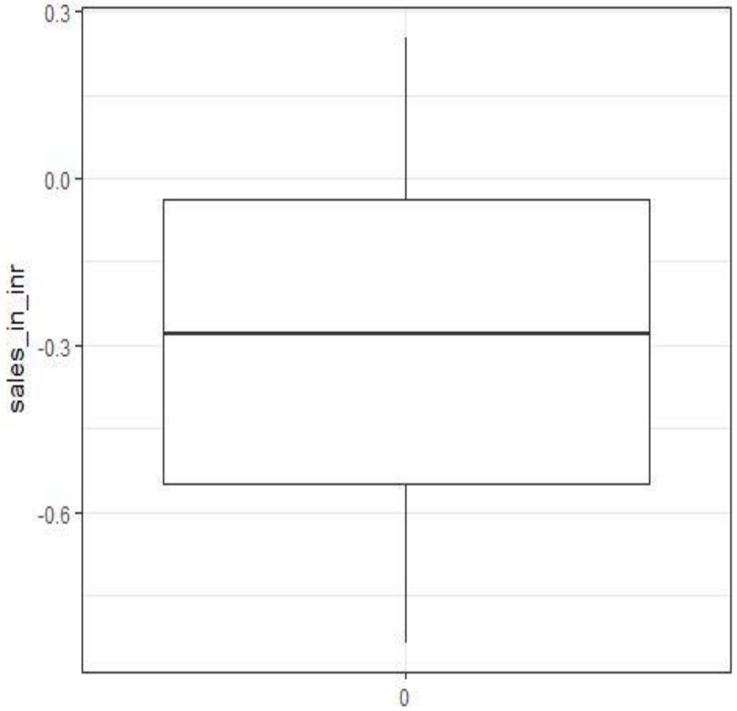
Handling outlier

Replacing outlier with average of its surrounding observations

Data with an outlier



Data without any outlier



Insights: The new data has no more outliers.

Fitting a linear regression model (no outliers present)

The simple linear regression model without any outlier results into the following –

Coefficients:

(Intercept)	rainfall_in_mm
-0.2904	-0.3598

RSS: 0.00413

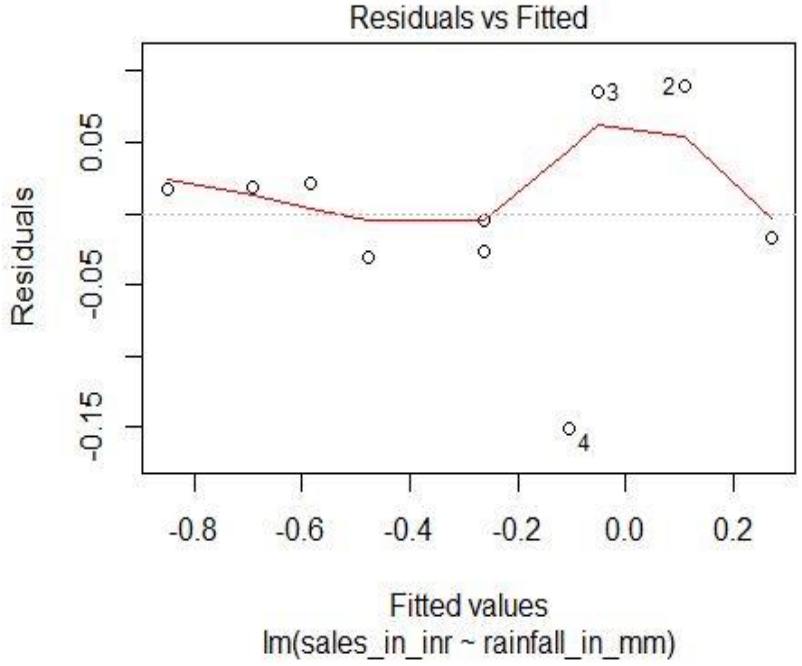
The simple linear regression equation becomes –

$$sales\_in\_inr = -0.2903623 - 0.3598104 * rainfall\_in\_mm$$

Insights: The given model has less RSS as compared with the model with an outlier.

Residual plot

- Residual plot comes handy when we need to verify if given model is suitable for dataset or not.
- If there isn't any significant pattern in the residual plot then model is acceptable.



Insights: There isn't any significant pattern seen in the curve. All the points are randomly scattered in the graph. Hence, given model is **acceptable**.

*— * — Hands On — * —*

Multiple Linear Regression

Consider we have been given with the actual values of a variable (say) X_p, then a multiple regression model response variable $\hat{y}$ is given as:

$$\hat{y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

Again as with simple linear regression model, the values of $\beta_p$can be calculated using *Gradient Descent* or *Coordinated Descent* methods.

Similarly, the best model is the one which has **least RSS** and other parameters like Adjusted R^2, AIC, BIC, etc. can also be used.

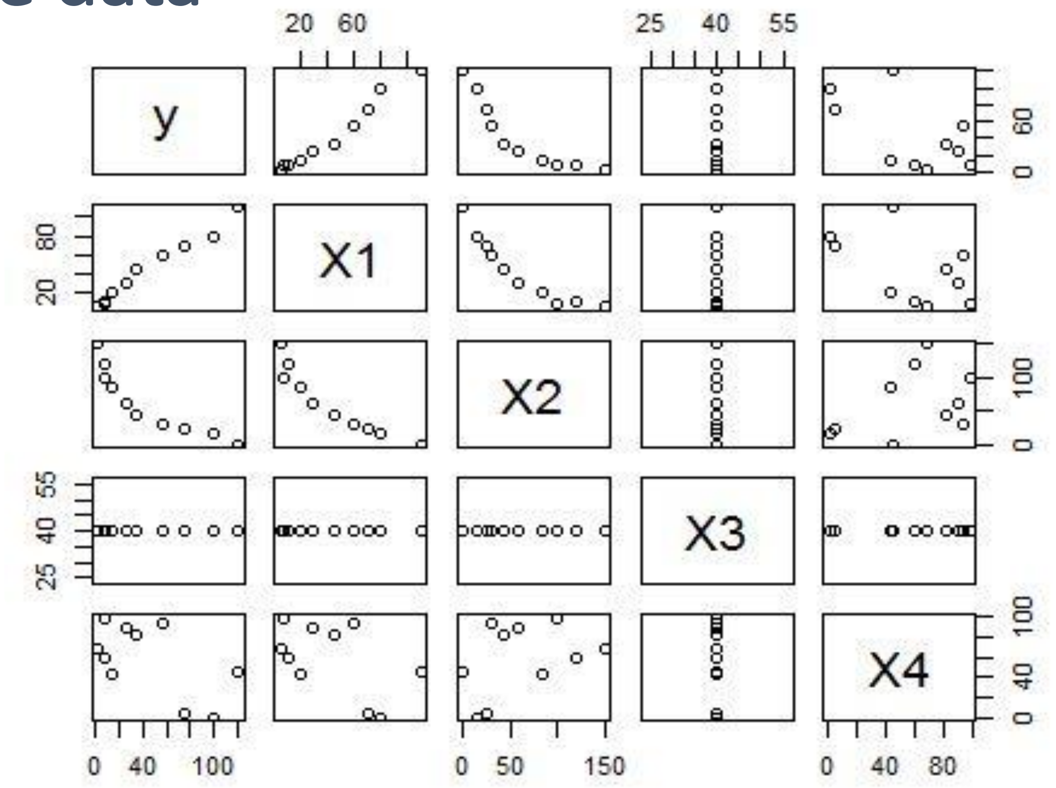
Multiple Linear Regression – Illustration

This time consider a new dataset with 4 independent variables and 1 dependent variable as shown below.

Our task is to determine y based on $X1, X2, X3$ and $X4$.

	y	X1	X2	X3	X4
1	2	5	150	40	68
2	8	12	120	40	60
3	9	8	100	40	98
4	15	20	85	40	43
5	26	30	60	40	89
6	34	45	44	40	81

Visualizing the data



Insights:

1. Response variable X1 is directly proportional to y
2. Response variable X2 is indirectly proportional to y
3. Response variable X3 is constant overtime and doesn't correlate to y
4. Response variable X4 has random nature. However, it has a little indirect relationship with y

Verifying relationship using covariance matrix

	y	X1	X2	X3	X4
y	1717.156	1442.444	-1813.44	0	-843.578
X1	**1442.444**	1244.222	-1598.89	0	-630.333
X2	-**1813.44**	-1598.89	2413.556	0	667.2222
X3	0	0	0	0	0
X4	**-843.578**	-630.333	667.2222	0	1207.789

Insights: The covariance matrix confirms the relationship of predictors to response variable.

Correlation is not causation

From the matrix we can even state that predictors are also related to one another. Therefore, out of one of the correlated predictors like (X1, X2), we can neglect one and build our model with another. <Procedure discussed ahead>

Note – If variables are correlated, it doesn't mean that one causes another.

Fitting a multiple regression model

A multiple linear regression model can be fit on the given dataset and following results are expected –

Coefficients:

(Intercept)	X1	X2	X3	X4
-5.0670	1.1846	0.0656	NA	-0.1164

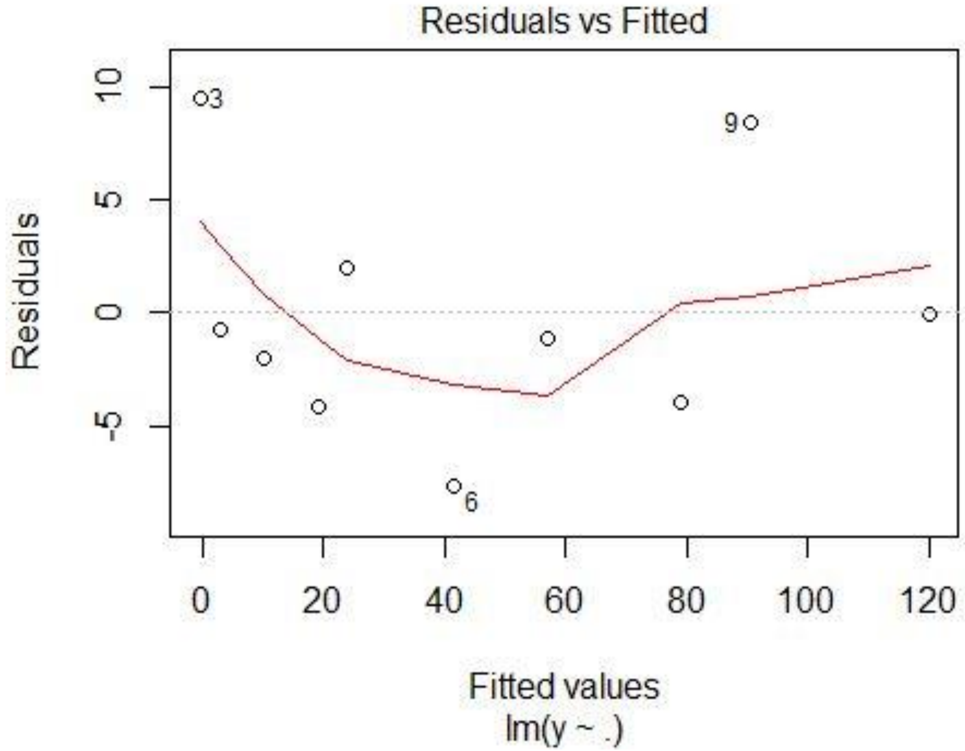
RSS: 26.22279

Model equation –

$$y \;= -5.07 + 1.18X1 + 0.06X2 - 0.11X4$$

Insights: We've build the model and it can be seen that a constant variable can be neglected.

Residual plot



Insights: There isn't any significant pattern seen in the curve. All the points are randomly scattered in the graph. Hence, given model is **acceptable**.

Variable Selection

Why?

- If correlated predictors are used while building a model, then model performance can be affected.
- More predictors increases complexity.

How?

There are various techniques to select most important variables when it comes to building a multiple linear regression model. We discuss subset selection methods –

1. **Forward Selection** – It *starts with just intercept* and keeps on adding the response variable one at a time which results in lower RSS until a certain threshold is reached. This approach can always be used.
2. **Backward Selection** – It *starts with all the predictors* and removes the ones with highest p-value to reach at a certain threshold. It is applicable only when the numbers of samples (n) are greater than the number of predictors (p).
3. **Mixed Selection** – It combines the added advantage of both the above mentioned approaches. Forward Selection uses greedy approach, hence early predictors becomes redundant. Henceforth, mixed selection remedy this bad effect.

Variable Selection – in practice

If we apply mixed selection on our dataset and build our multiple regression model we attain following results –

Coefficients:

(Intercept)	X1	X4
3.624	1.095	-0.127

RSS: 27.52

Insights:

From the above results we can infer that variable X2 can also be neglected. Last time RSS = 26 and this time RSS = 27. Hence, we retained the approx. RSS but with increased performance.

$- * - \textit{\textbf{Hands On}} - * -$

Polynomial Regression

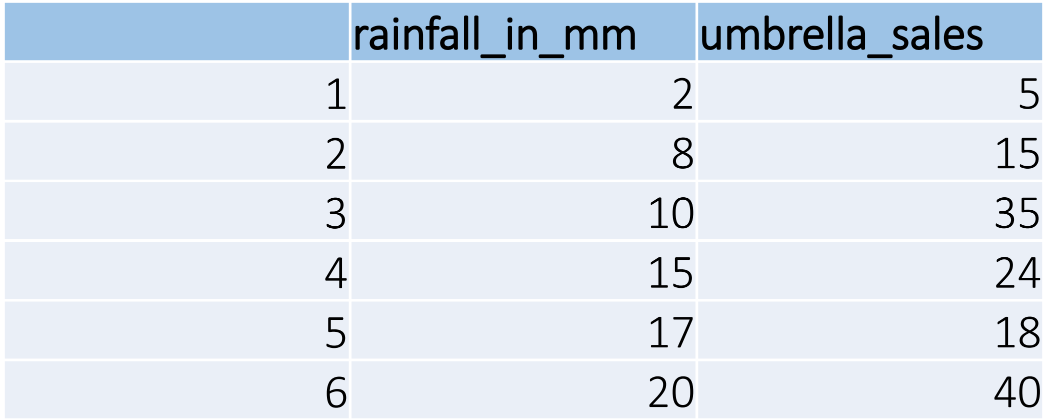
The last two regression models tries to fit a straight line instead of a curve. However, polynomial regression does the reverse, it fits a polynomial of certain order to estimate response variable.

Let’s say if there is only one predictor and we are trying to fit a polynomial of degree 10. Then the resultant model will consists of 10 β values particular to that predictor. Similarly, if there are p predictors then for each of the predictor there would be 10 β values .

Polynomial Regression – Illustration

Consider a reformed rainfall vs sales dataset which has similar scale to one another as shown below.

We need to form a model on given dataset and this time even need to **predict** sales with new values of rainfall attribute.



	rainfall_in_mm	umbrella_sales
1	2	5
2	8	15
3	10	35
4	15	24
5	17	18
6	20	40

Training and Test Data

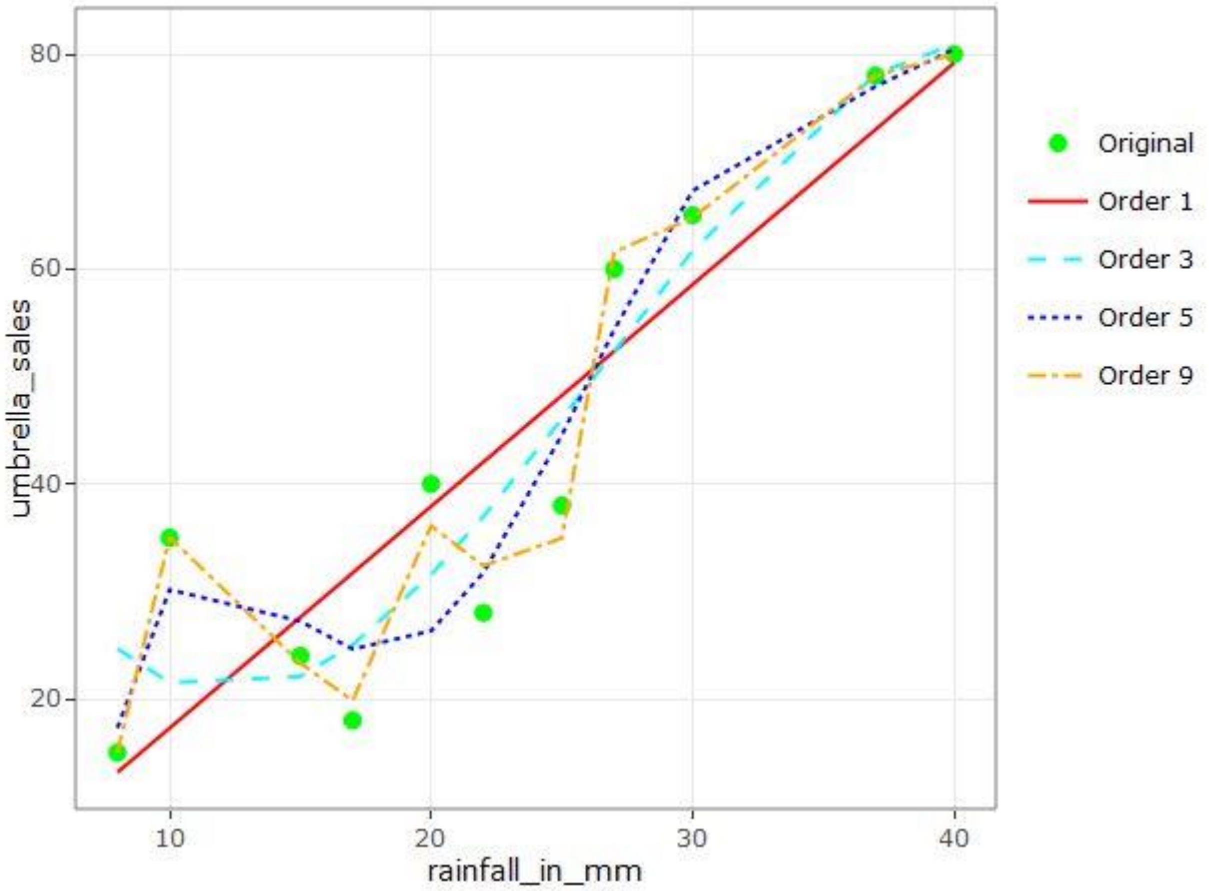
- To predict new sales values, we need some unused rainfall instances which are not provided during model formation.

- Therefore, we split out original dataset into two unequal parts –
  1. **Training data** – consists of 75% of the original data
  2. **Test data** – consists of 25% of the original data

- Assigning of values to each parts is performed **randomly** so that both the data has points from every part of the dataset.

Note – It's not mandatory to follow the ratio of 75:25.

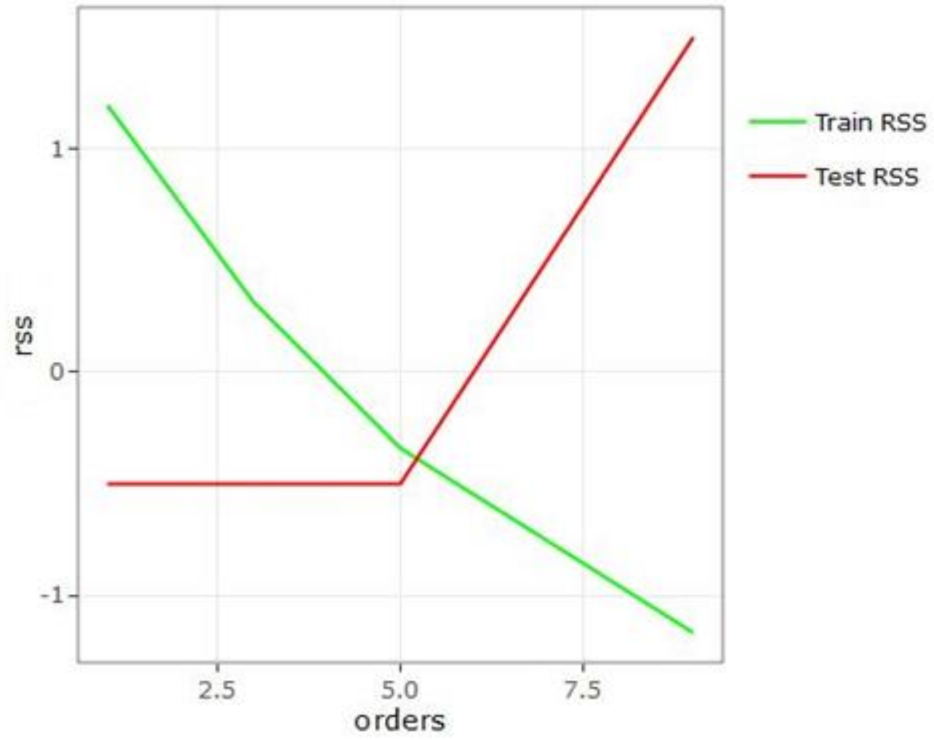
Fitting polynomial regression model and visualizing the fitted model



Insights: Order 1 – under fits the model
Order 9 – over fits the model

Comparing Train and Test RSS

The test RSS is calculated using the test data output.



Insights:

As the order increases, it leads to overfitting and –

- Train RSS decreases
- Test RSS increases

Choosing best value for degree of polynomial

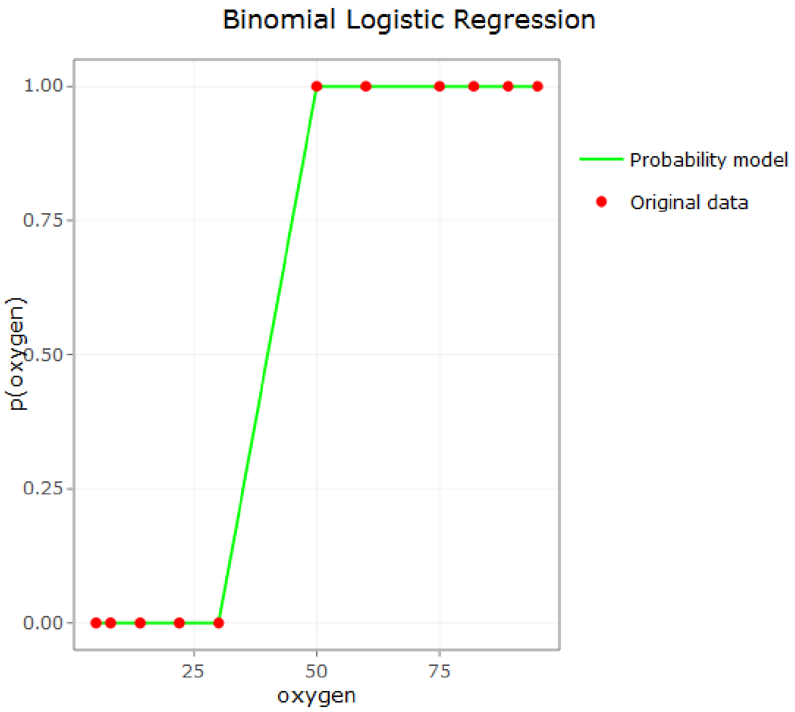
The best value of polynomial degree can be determined using Regularization methods (to be discussed later) like –

- Ridge Regression
- Lasso Regression
- Elastic-Net Regression

$-\ * - \boldsymbol{Hands\ On} - * -$

Logistic Regression

- Logistic Regression is a supervised classification machine learning approach yet it goes with a suffix ***<u>Regression</u>*** because for a given input it gives us the probability of the outcome category.
- We discuss two types of logistic regression namely –
  - Binary (two categories)
  - Multinomial (more than two categories)



Terminologies

$\boldsymbol{Odds}$ $\boldsymbol{\frac{p}{1-p}}$	Ratio of success to failure. For instance, if, $P(success) = 5/6$ $P(failure) = 1/6$ $Odds = \frac{5/6}{1/6} = 5/1$	$[0, \infty]$
$\boldsymbol{Logit}$ $\boldsymbol{l = \ln\left(\frac{p}{1-p}\right)}$	Logit of Odds or Log Odds	$[-\infty, \infty]$ Log odds is used in spite of Odds due to its wide range of possible outcomes
$\boldsymbol{Logistic\ transformation}$ $\boldsymbol{p = \frac{e^l}{1+e^l}}$	Inverse of logit -> Logistic regression model. $\ln\left(\frac{p}{1-p}\right)$ $= \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$ $p = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_p X_p}}$	$[0,1]$

Binary Logistic Regression – Illustration

Let us understand the logistic regression when we have two categorical outcomes.

Consider a case of an astronaut who carries oxygen cylinders with him. A dataset has been provided, and we need to predict if an astronaut survives or not (response variable) based on the amount of oxygen (predictor variable) remains in his cylinder.

	oxygen	survive
1	10	0
2	5	0
3	75	1
4	14	0
5	8	0
6	50	1

Fitting binary logistic regression model

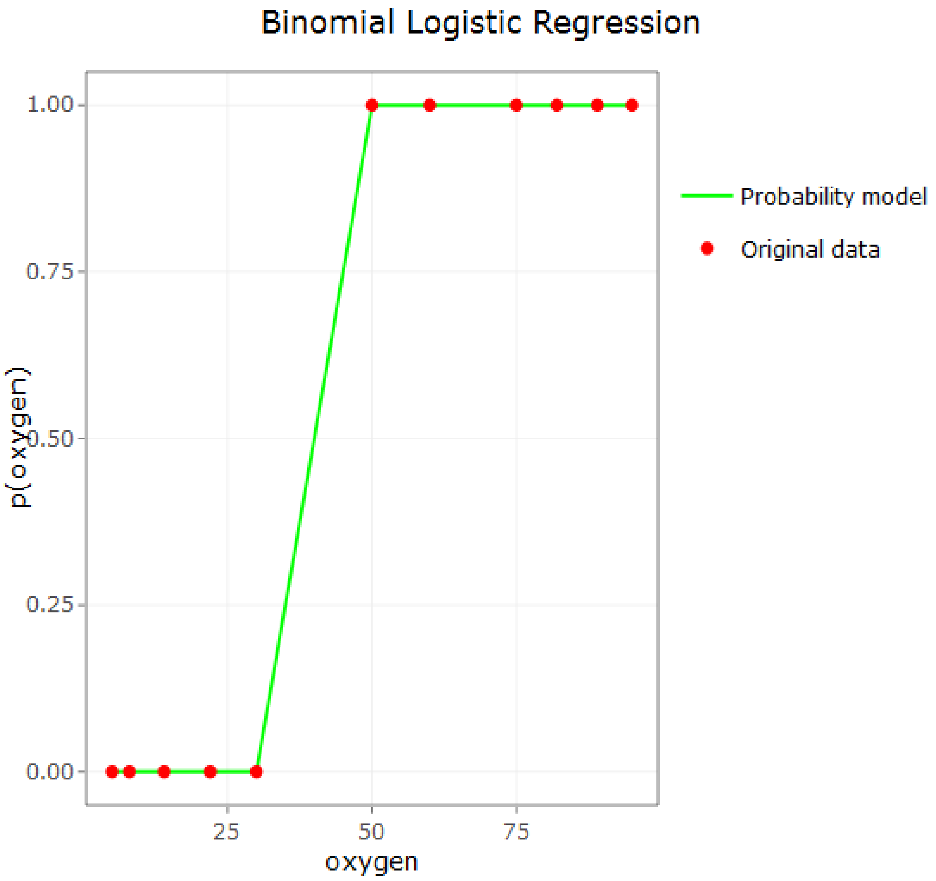
We converted data into training and testing data followed with building of model. The expected results are shown below –

Coefficients:

(Intercept)	oxygen
-92.173	2.303

Insights: From the model we can state that for a unit change in "oxygen", log odds change by 230.3%

Visualizing the fitted model



Insights: The green line (can be made curve for better illustration) depicts the probability likelihood for oxygen data.

Threshold probability & Accuracy of the model

- From the fitted model, we can obtain probabilities for the test data.
- Next step is to assign a threshold probability using which, categories will be assigned based on where does a given probability lies (before or after threshold).
- Choosing a threshold probability lies in the hands of domain expert.
- Once done we can find the accuracy of the model using confusion matrix

	predicted		
actual	0	1	Sum
0	2	1	3
1	0	1	1
Sum	2	2	4

Insights:

The confusion matrix shows that one value (red) has been misclassified under 1 (in prediction). Three values have been classified correctly (green). The accuracy of the model = Sum(green)/blue = **0.75**

Multinomial Logistic Regression

- Using binary LR, we could only classify 2 categories. To extend the need for more than two categories we rely upon multinomial LR.
- It works by finding the log odds of two categories and considering the left out as a **reference class**.
- Consider having a car dataset consisting of 3 different cars category namely *Toyota, Ferrari* and *Bugatti* and corresponding to each one of them is their listed budget.
- Let us assume "Ferrari" to be as a reference category, then other 2 logit models can be formed as –

$$ln\frac{p(outcome = Toyota)}{p(outcome = Ferrari)} = \beta_0^{Toyota} + \beta_1^{Toyota} * Budget = l_{Toyota}$$

$$ln\frac{p(outcome = Bugatti)}{p(outcome = Ferrari)} = \beta_0^{Bugatti} + \beta_1^{Bugatti} * Budget = l_{Bugatti}$$

Multinomial Logistic Regression (Contd.)

- Also, summation of all probabilities is equal to 1. So,

$$p(outcome = Ferrari) = 1 - p(outcome = Toyota) - p(outcome = Bugatti)$$

- Solving above three equations provide the probability of all three classes as shown below –

$$p(outcome = Toyota) = \frac{l_{Toyota}}{1 + l_{Toyota} + l_{Bugatti}}$$

$$p(outcome = Bugatti) = \frac{l_{Bugatti}}{1 + l_{Toyota} + l_{Bugatti}}$$

$$p(outcome = Ferrari) = \frac{1}{1 + l_{Toyota} + l_{Bugatti}}$$

Multinomial Logistic Regression – Illustration

- A sample of dataset is shown.
- We set Ferrari as reference variable.
- We make train and test data.
- Next, we build multinomial logistic regression model.

	budget	cars
1	100000	Toyota
2	250000	Toyota
3	500000	Toyota
4	750000	Toyota
5	1000000	Toyota
6	2000000	Ferrari

Coefficients:

```
          (Intercept)        budget
Bugatti   -25.66995          3.595444e-06
Toyota    20.08826           -1.361525e-05
```

Insights: From above we can build logit models for *Bugatti* as well as *Toyota*. Using them we can get the probability of our reference class (*Ferrari*) too.

Predicting and finding accuracy of the model

We can predict the classes using test data and then compare the predicted classes with actual classes to gain insights upon accuracy of the model.

The confusion matrix is shown below –

	Predicted			
Actual	Ferrari	Bugatti	Toyota	Sum
Ferrari	1	0	0	1
Bugatti	0	2	0	2
Toyota	0	0	2	2
Sum	1	2	2	5

Insights: The given model has 100% accuracy on the given dataset with no misclassification.

k – Nearest Neighbors

- $k - Nearest\ Neighbors$ is another type of supervised machine learning algorithm which classifies data based on its k number of nearest data points.

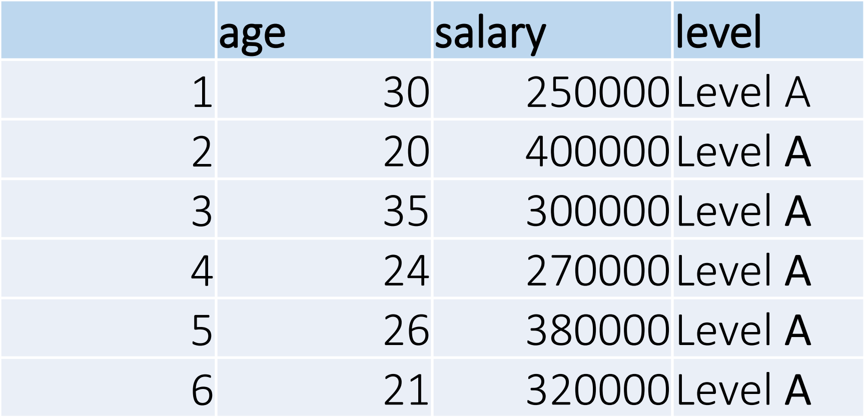
- kNN uses **distance** as a metric to classify a data point to a category.
  1. If the predictors are numeric then we can use *Euclidean distance*.
  2. If predictors are categorical then we have to rely upon *Hamming distance*.
     (Hd = Number of ones in A xor B)

Note – We constraint our discussion only to Euclidean distance.

k – Nearest Neighbors – Illustration

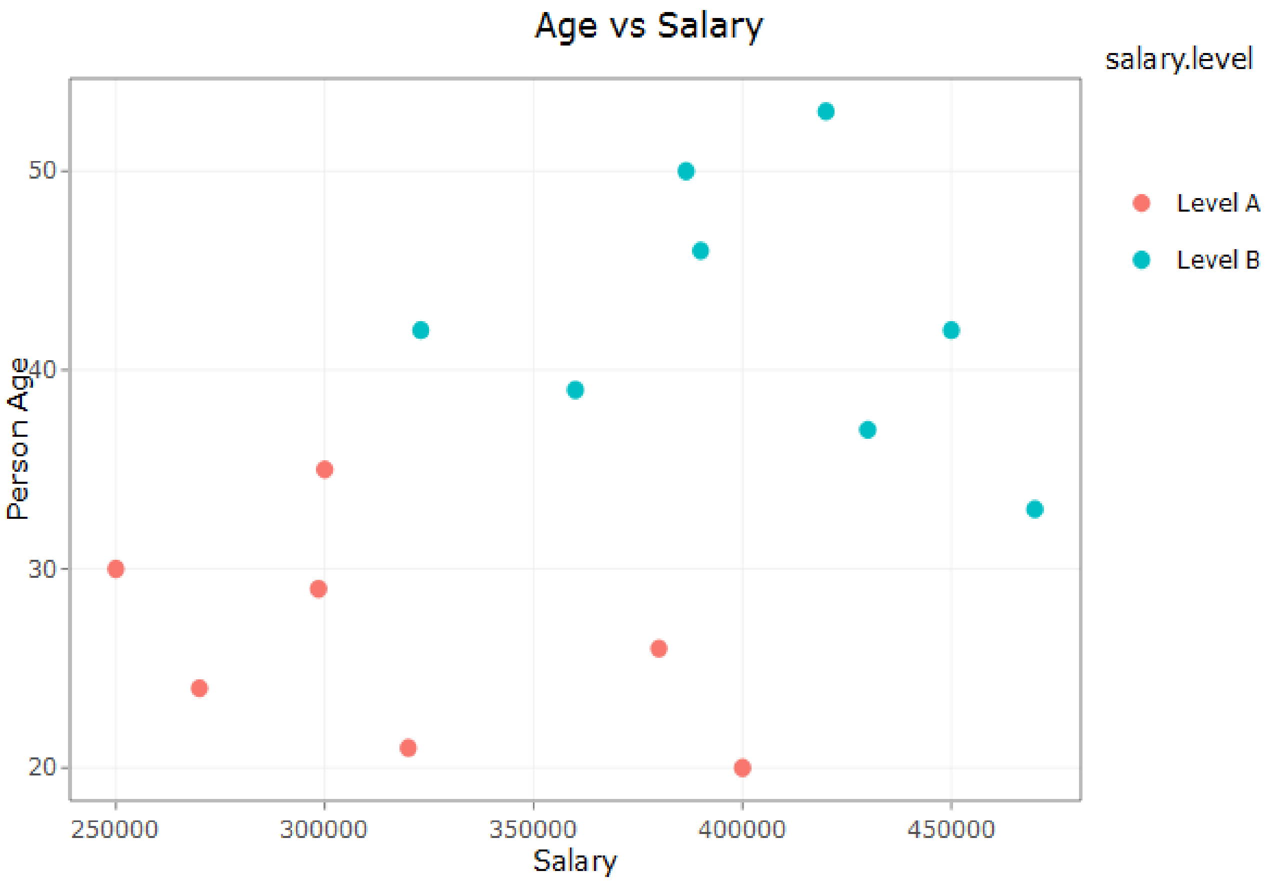
Consider a dataset with 2 independent variables namely “age” and “salary” and a dependent variable namely “level” consisting of two levels A and B.

We need to classify a person into either level based on his age and salary.



	age	salary	level
1	30	250000	Level A
2	20	400000	Level A
3	35	300000	Level A
4	24	270000	Level A
5	26	380000	Level A
6	21	320000	Level A

Visualizing the data



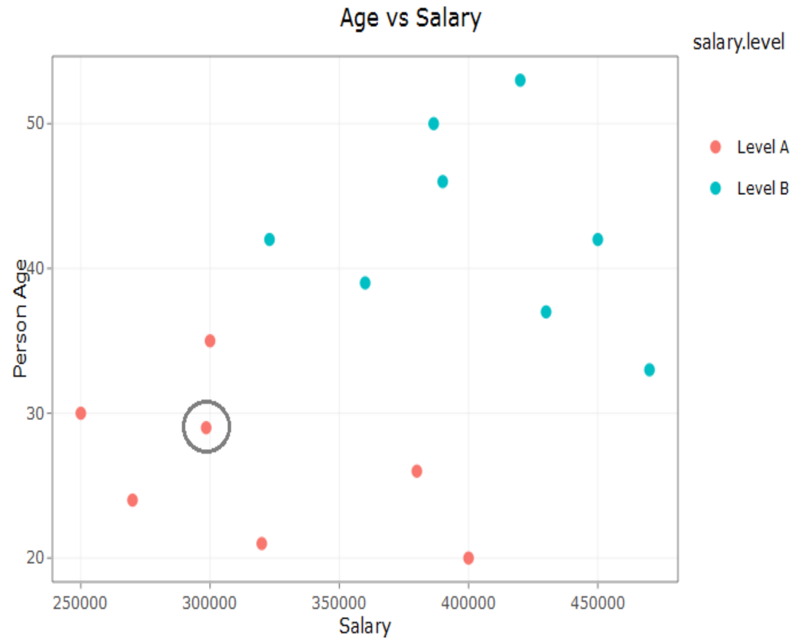
Insights: Salary scale is higher than Age scale

Distance calculations

Now, let us try to understand how distance is used given $k = 1$.

Consider the encircled data point from the figure.

We need to find out its distance from all the data points and observe to which closer level (A or B) data point it belongs to.



To recall what a Euclidean distance is, consider two coordinate points (x_1, y_1) and (x_2, y_2) then,

$$Euclidean\ distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Normalization of scales

As observed during visualization that both predictors scale differ.

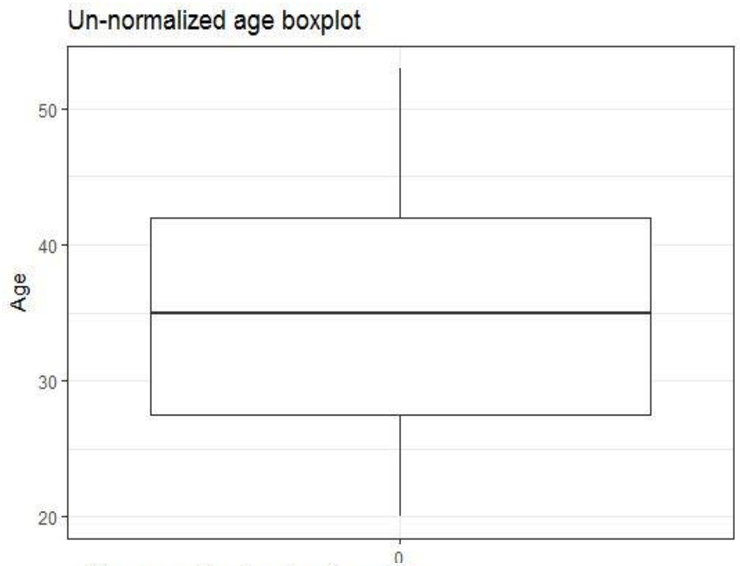
We can use min-max normalization which results into a range $[0, 1]$.

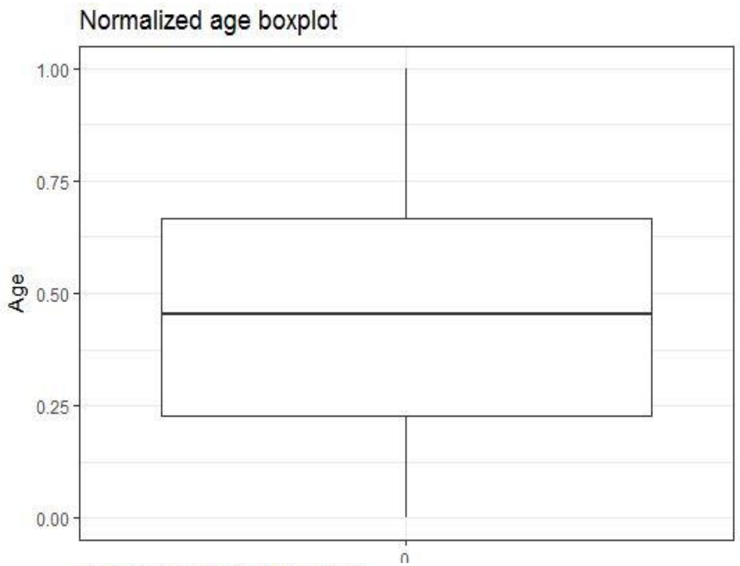
$$\min - \max normalization = \frac{x - \min(x)}{\max(x) - \min(x)}$$

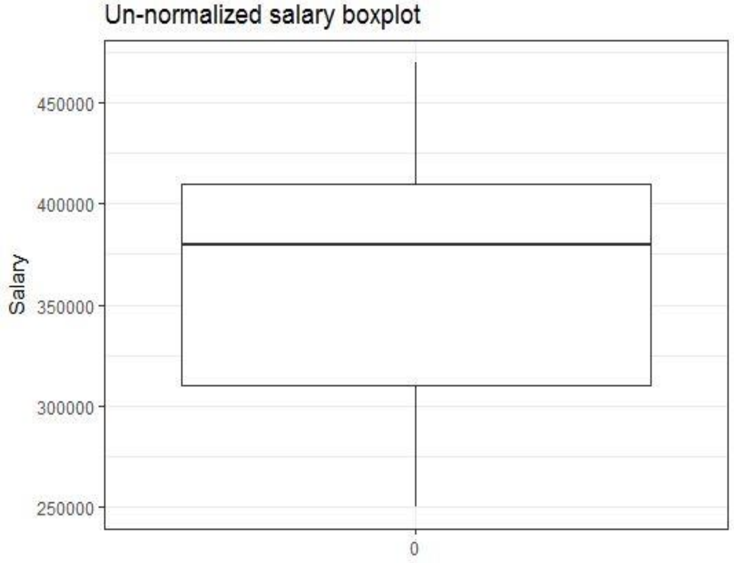
Let us try to compare the Euclidean distance (ED) of chosen encircled data from few data points with and without normalization.

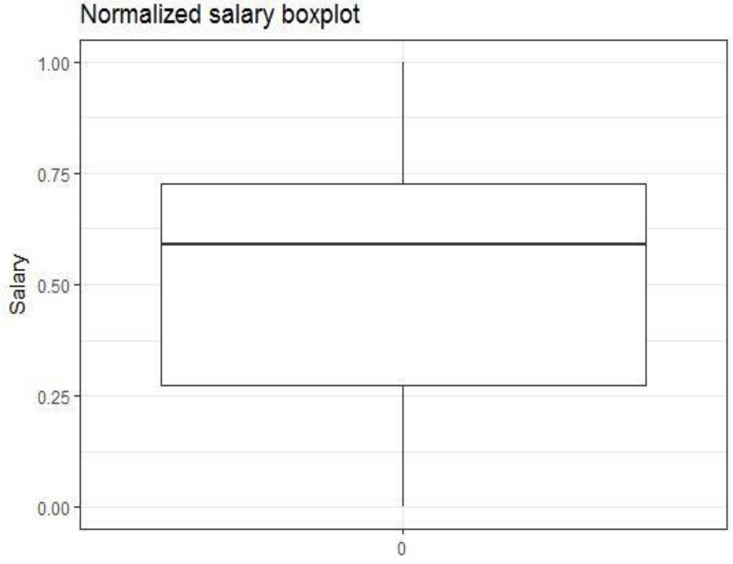
Un-normalized data	Normalized data	Comment
[1] "ED = 48500"	[1] "ED = 0.22"	
[1] "ED = 101500"	[1] "ED = 0.54"	
[1] "ED = 1500.01"	[1] "ED = 0.18"	Both values show this is closest
[1] "ED = 28500"	[1] "ED = 0.2"	Norm. value shows it is second closest whereas Un-norm value conflict with this result. This shows why norm. is necessary.
[1] "ED = 81500"	[1] "ED = 0.38"	
[1] "ED = 21500"	[1] "ED = 0.26"	

Verifying structure of the normalized data

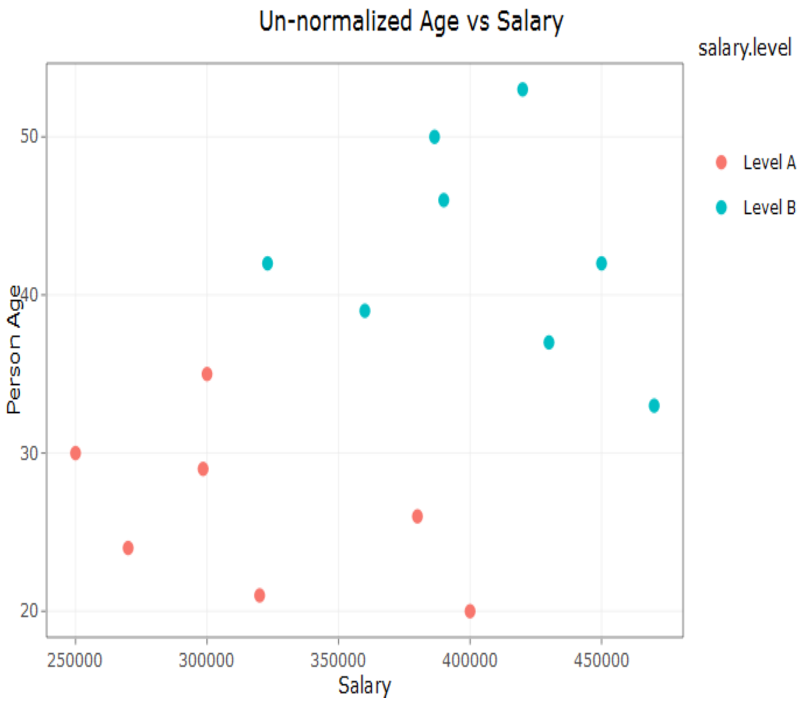


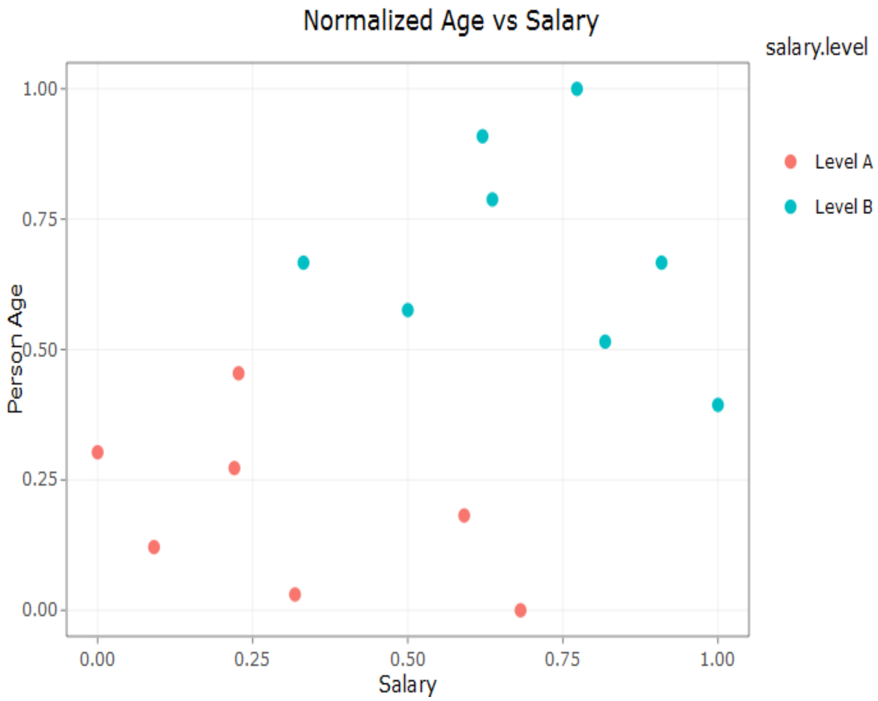






Verifying structure of the normalized data





Insights: Data structure remains same even after normalization

Fitting kNN model and finding accuracy

- Split original data into train and test data.
- Choose value of k (say $k = 1$).
- Build kNN model.
- Find accuracy using confusion matrix.

	Predicted		
Actual	Level A	Level B	Sum
Level A	2	0	2
Level B	0	3	3
Sum	2	3	5

Insights: Given data is 100% accurate.

Finding best value of k

Best value of k is the one which results into higher accuracy relative to others.

There are two ways to achieve this –

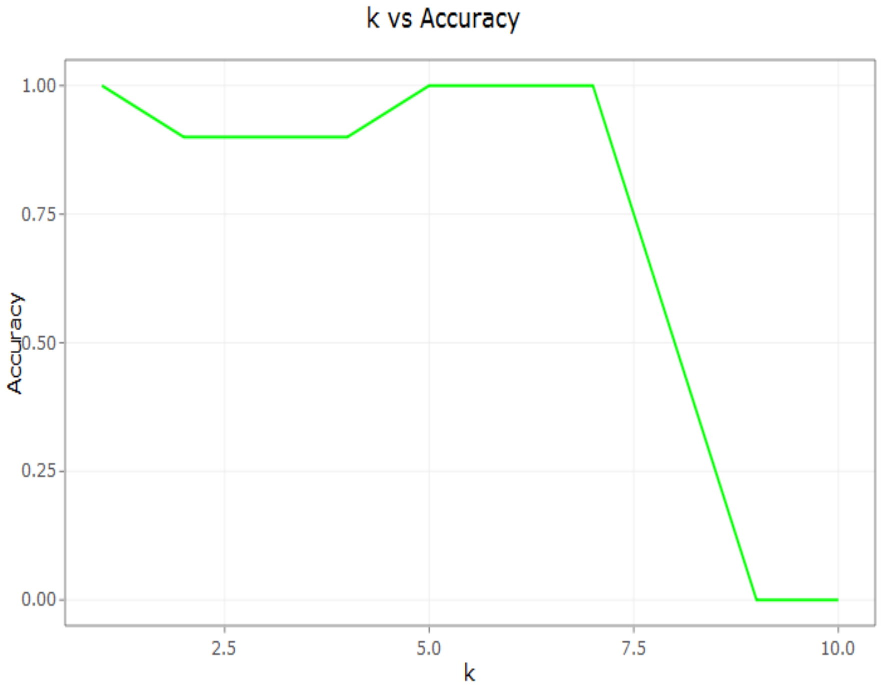
- **Manual** – Passing different values of k and choosing the one which results into higher accuracy.
- **k fold Cross – validation** - k fold CV randomly divides given training data into k equal subparts. It uses k-1 subparts for training and k^{th} subpart for testing. For each subparts it calculates the accuracy on provided range of k. The k results from the folds can then be averaged to produce a single estimation.

k fold CV

Consider a training data broken into 5 subsets i.e. we are using 5 fold CV–

Set 1	Set 2	Set 3	Set 4	Set 5

Iteration	Training subset	Testing subset	Accuracy
1	1, 2, 3, 4	5	80
2	2, 3, 4, 5	1	75
3	1, 2, 4, 5	3	60
4	1, 3, 4, 5	2	50
5	1, 2, 3, 5	4	78



Insights: The k results from the folds are averaged to produce a single estimation.

– * – R Hands On – * –

Support Vector Machines

- SVM is used to classify data points into categories based on some past training model.
- The task of the SVM is to define an **optimal hyperplane** which separates one category from another.
- The hyperplane is always *one less* than the number of dimensions.
- SVM predicts class using LDA (Linear Discriminant Analysis), function of whose is given by –

$$f(x) = \sum w_i^T x_i + b$$

x →training or test data
w →weight vector, normal to the hyperplane
b →bias

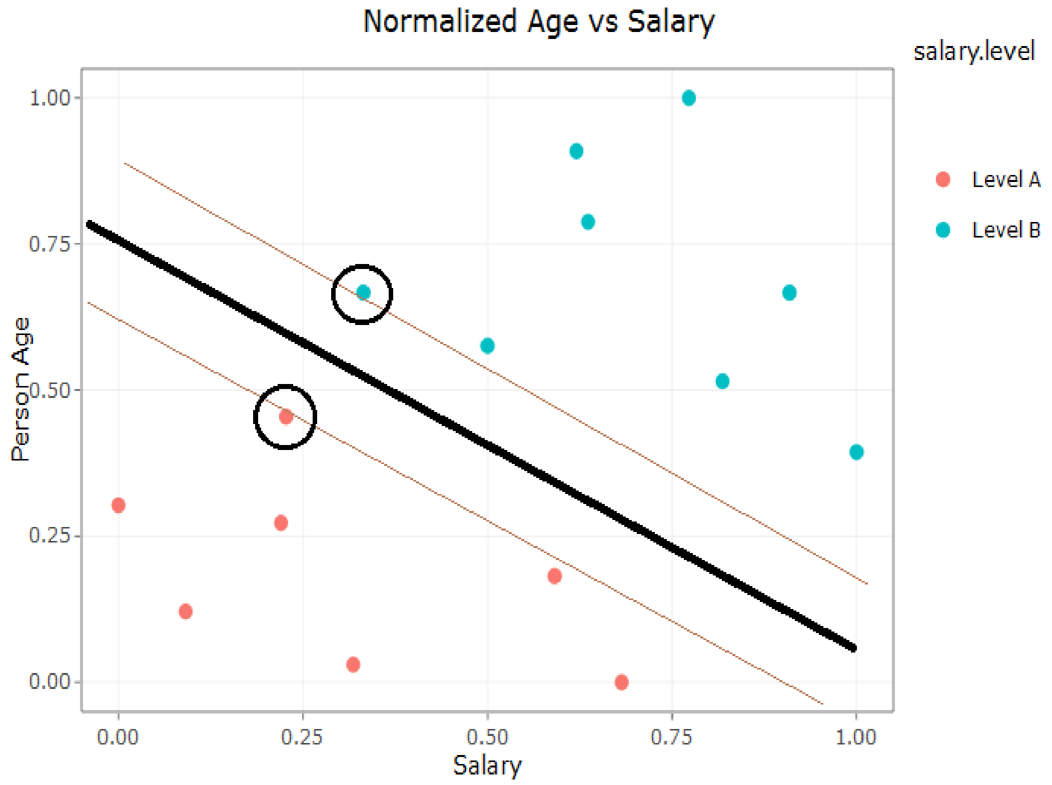
The above equation can be broken into the form of $y \ = \ mx + c$ as –

$$w = \begin{pmatrix} 1 \\ m \\ c \end{pmatrix} \ and \ x = \begin{pmatrix} y \\ x \\ 1 \end{pmatrix}$$

The term $w_i^T x_i$ represents the dot product.

Terminologies

Bold black line	Optimum hyperplane
Brown lines	Margin lines; occurs at equal distance from hyperplane. Distance between these 2 brown lines is called as margin.
Encircled points	Support Vectors; points falling on margin lines.

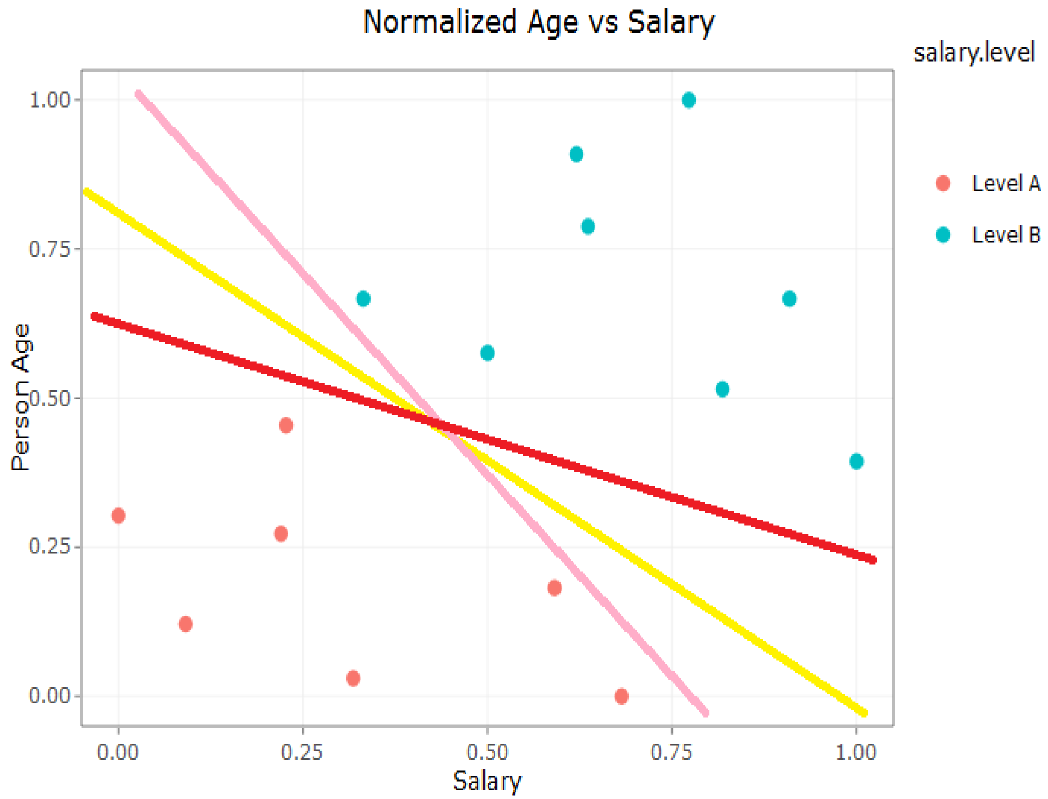


Optimum hyperplane

Consider previous normalized age and salary dataset to predict salary level.

In the figure given here, three lines (yellow, pink and red) have been drawn.

It the **objective of SVM** to find such optimum best fit line which helps to separate the categories but also comes handy to classify new data points correctly.



Lines pink and red are very close to each category and therefore may misclassify new data. Here, yellow line is best compared with other two.

SVM – Illustration

- Let us try to classify our previous age, salary and level dataset using SVM.
- Build the SVM model. The expected output consists of the following –

Parameters:

SVM-Type: C-classification

SVM-Kernel: linear

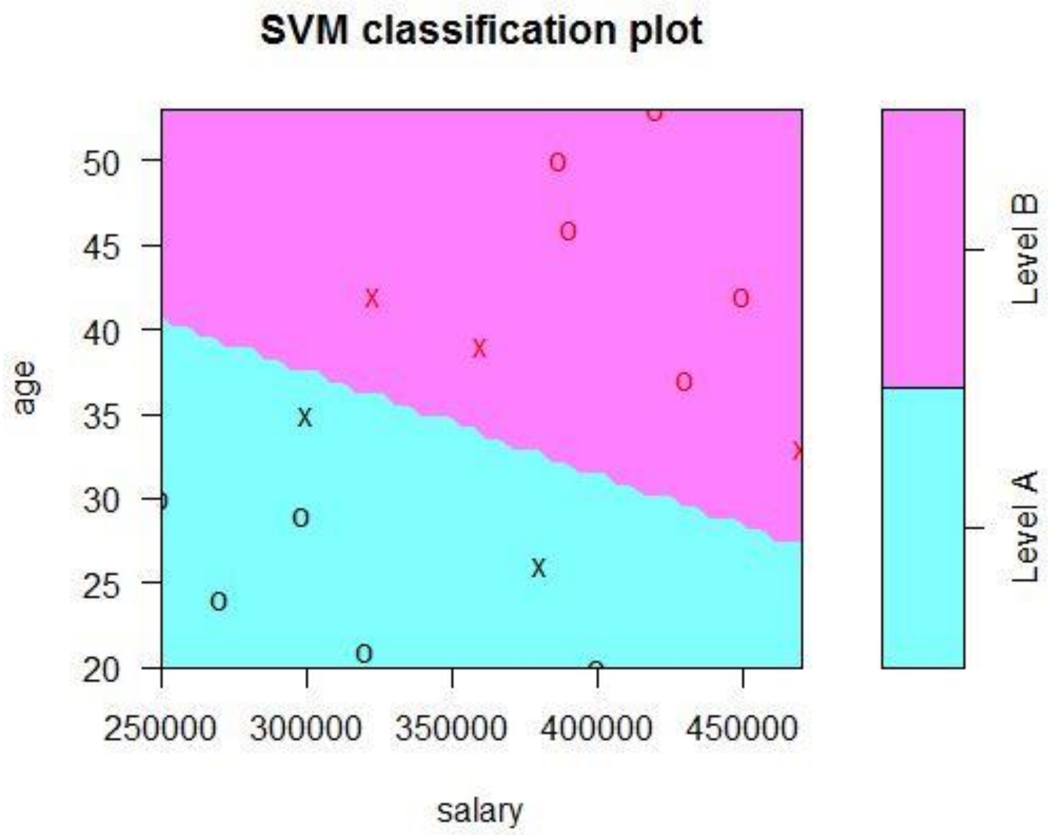
cost: 1

gamma: 0.5

Number of Support Vectors: 5

Insights: There are 5 data points (support vectors) on margin lines. The kernel used to separate the categories is **linear**. We will discuss the relevance of cost and gamma soon.

Visualizing the fitted model



Insights: The points depicted using (**X**) signifies the Support Vectors.

Hard and Soft Margins Classifiers

Hard Margin Classifier – If all the data points from all the categories irrespective of where they occur are clubbed together, then such SVM is known to use hard margin classifier. Such classifier tends to show overfitting.

Soft Margin Classifier – If a margin is traced among the categories even leaving few data points misclassified then such SVM is known to use soft margin classifier.

A term “**Slack Variable**” represented by ε is introduced inside soft margin classifier which defines the distance of misclassified point from its class margin.

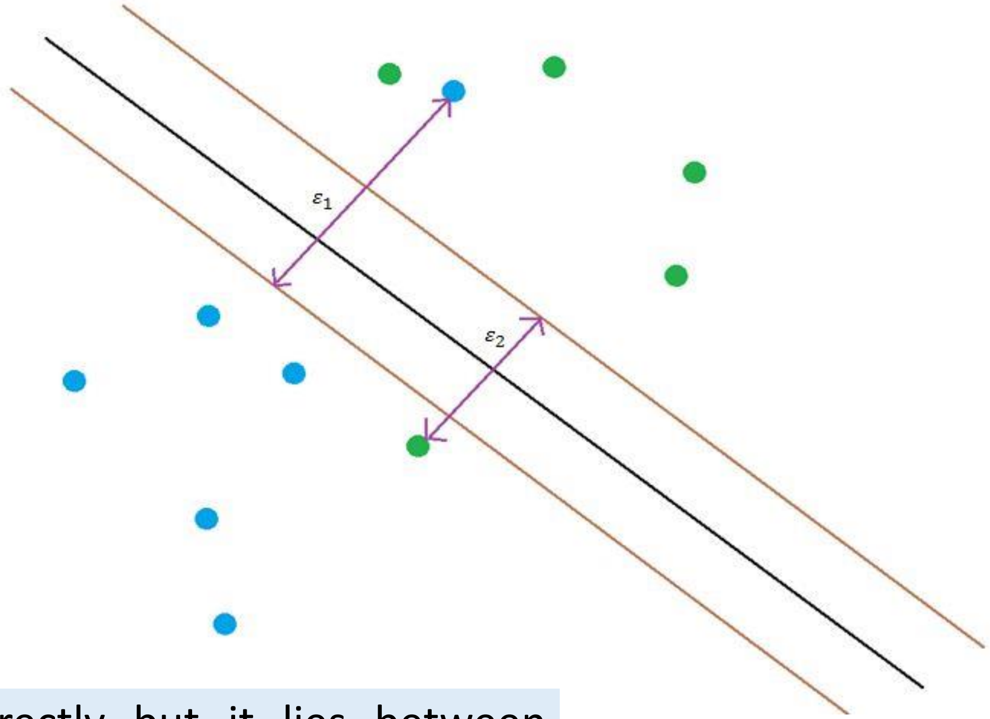
Slack Variable ε

Here one green and one blue point has been misclassified.

ε_1 gives the distance of blue point from its class margin line

And,

ε_2 gives the distance of green point from its class margin line.



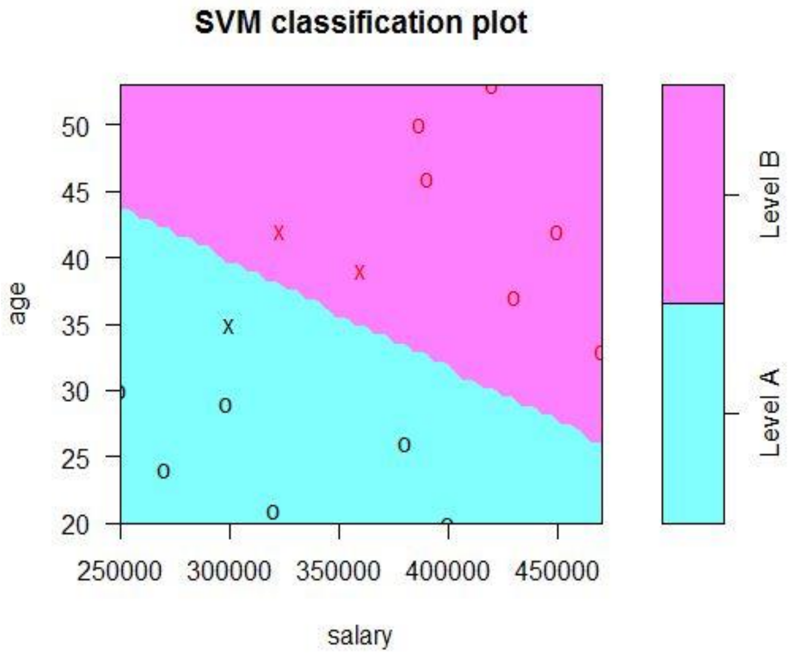
$\mathbf{0 \leq \boldsymbol{\varepsilon} \leq 1}$	Point classified correctly but it lies between hyperplane and its class margin line. Hence, does margin violation.
$\boldsymbol{\varepsilon} > \mathbf{1}$	Point is misclassified and lies beyond margin line.

Regularization parameter or Cost

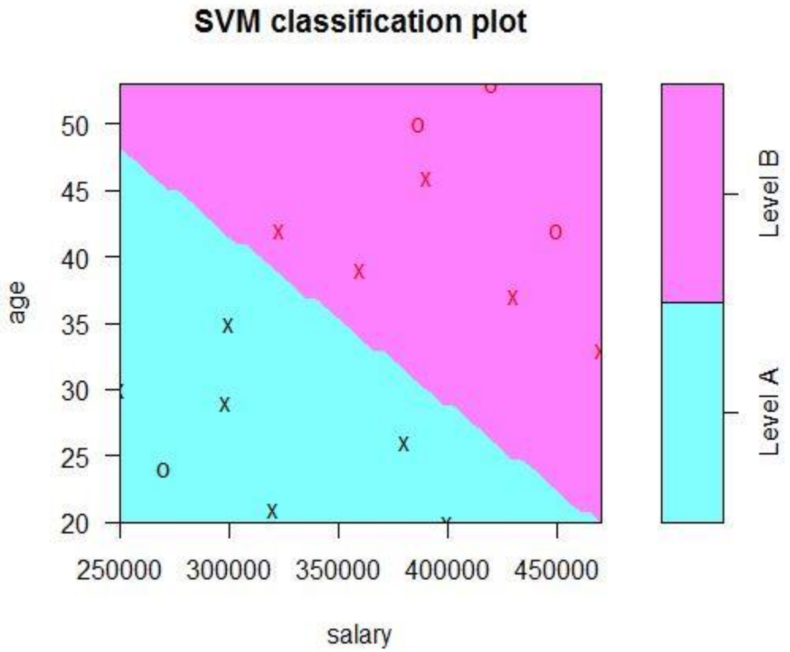
- The value of C is inversely proportional to the margin length.
- If C is small then margin is large (case of soft margin classifier).
- If C tends to approach infinity then margin is too biased on data (case of hard margin classifier). Our model has C = 1, which is acceptable.

Comparing SVM models on different C

$C = 10000$



$C = 0.1$



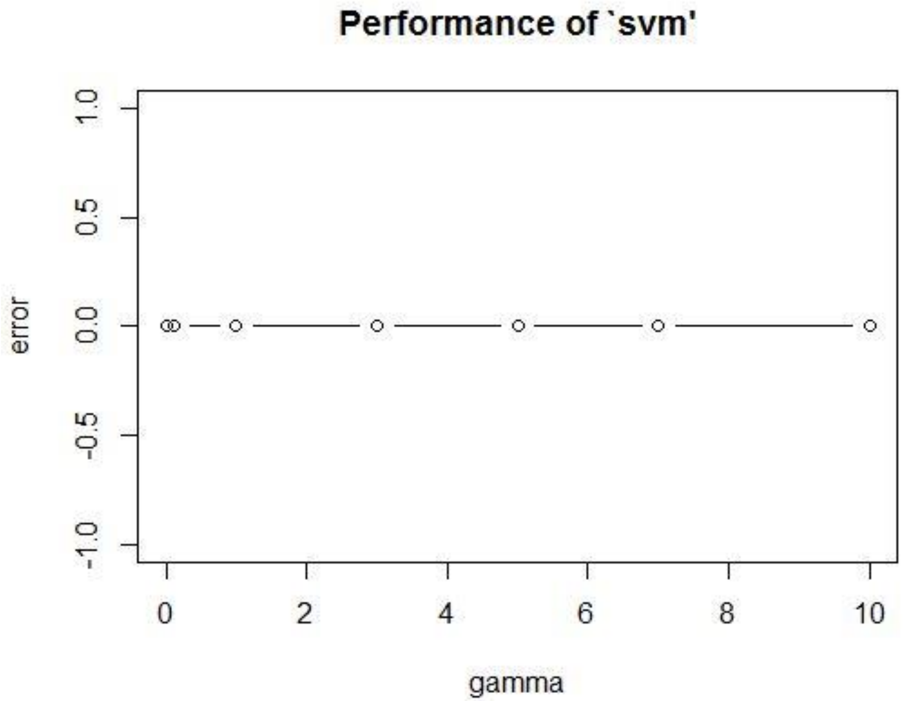
Insights:

- Left plot has large value of C and hence we can observe that margin length has been shrunk (only 3 Support Vectors (X)).
- On the other hand, we have smaller value of C and hence margin length is expanded (10 Support Vectors (X)).
- CV can be used to find best value of C.

Hyper parameter gamma

- Gamma is a hyper-parameter used to estimate the **performance of SVM**.
- The best value of gamma can also be found out using CV.

The CV results of gamma for our dataset is as follow –



Parameters:

SVM-Type: C-classification

SVM-Kernel: linear

cost: 1

gamma: 0.001

Number of Support Vectors: 5

Insights: No misclassified value on all the provided values to gamma

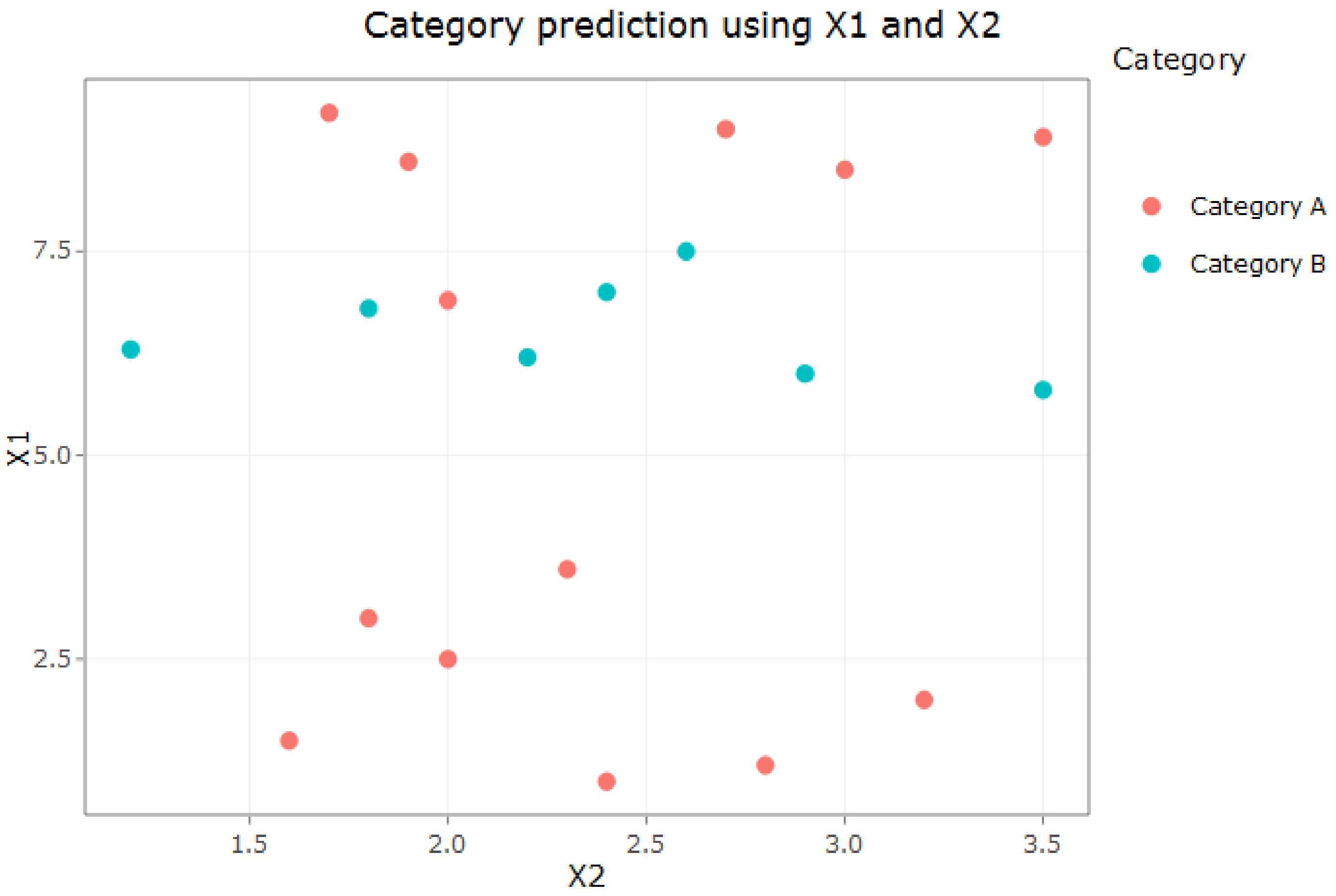
— * — R Hands On — * —

SVM for non-linear kernel

Consider a simulated dataset with 2 predictors $X1$ and $X2$ with a response variable y having two categorical outcome A and B.

	X1	X2	y
1	1	2.4	Category A
2	1.2	2.8	Category A
3	1.5	1.6	Category A
4	2	3.2	Category A
5	2.5	2	Category A
6	3	1.8	Category A

Visualizing the data



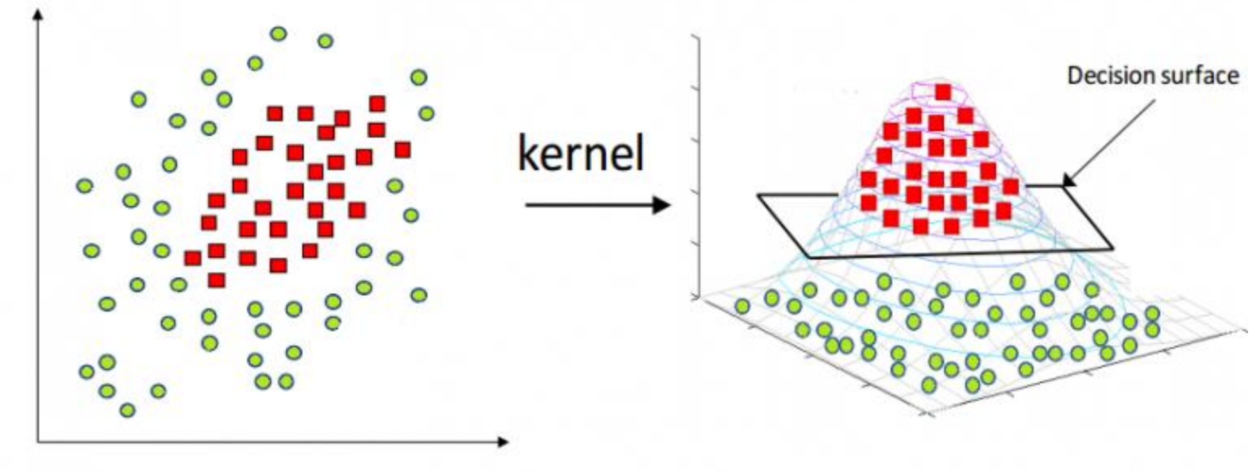
Insights:

- Category A is widespread in the scatter plot and hence can’t be separated with a linear classifier.
- Both scales are in similar range to each other.

Kernel Trick

The concept of kernel trick is that it **transforms the data into higher dimension** then the existing one and then classifies data.

Consider graph shown below retrieved from here –



Here, red points have been taken into third dimension and then classification is performed by building a hyperplane of dimension 2.

Fitting polynomial kernel

- We can find the degree of best polynomial using CV.
- With the best degree polynomial we can build our non-linear kernel SVM classifier.
- The expected model parameters –

Parameters:

SVM-Type: C-classification

SVM-Kernel: polynomial

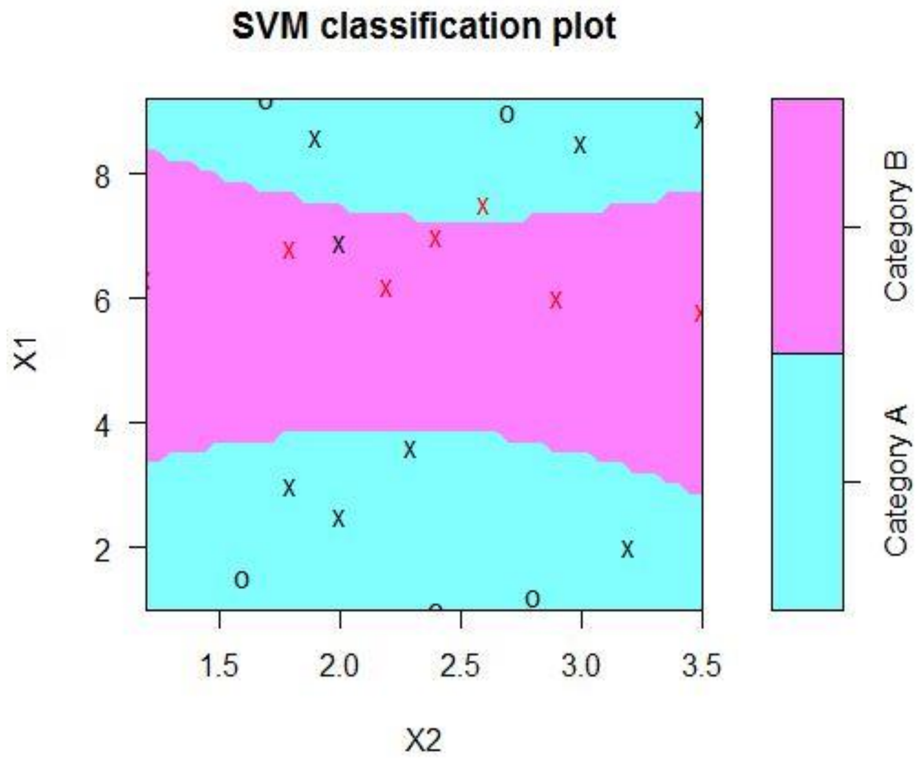
cost: 1

degree: 2

gamma: 0.5

coef.0: 0

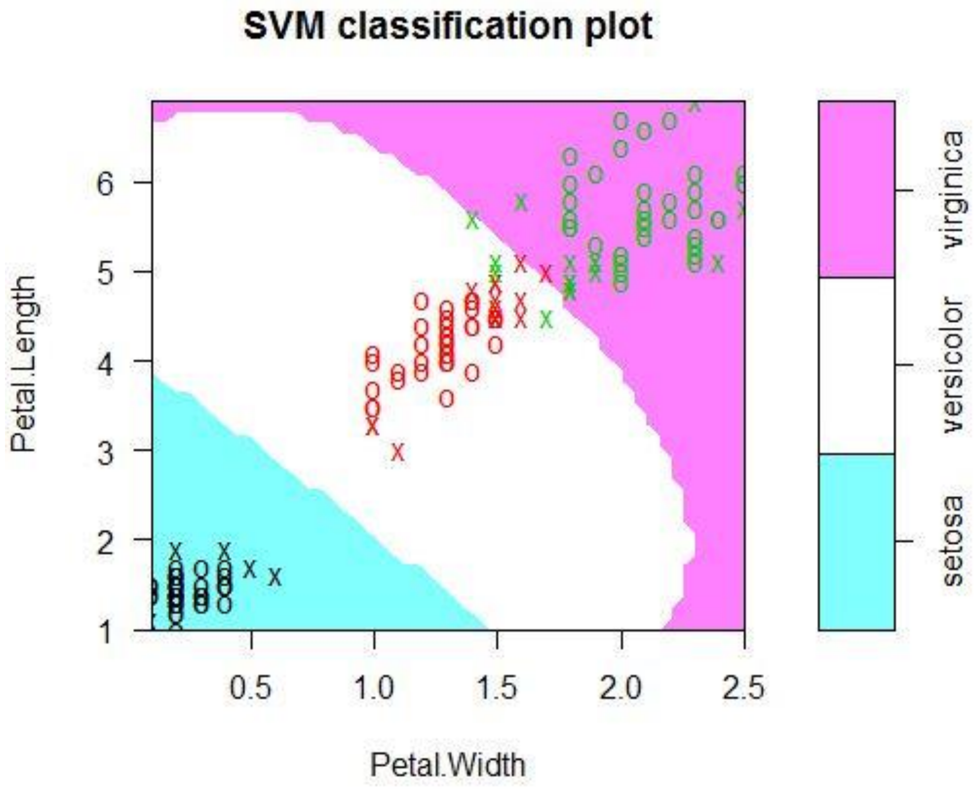
Number of Support Vectors: 15



– ∗ – R Hands On – ∗ –

Multi class SVM

Consider iris dataset with only Petal attributes and target as Species. Deploying SVM with radial kernel we can classify data into all 3 categories as shown –



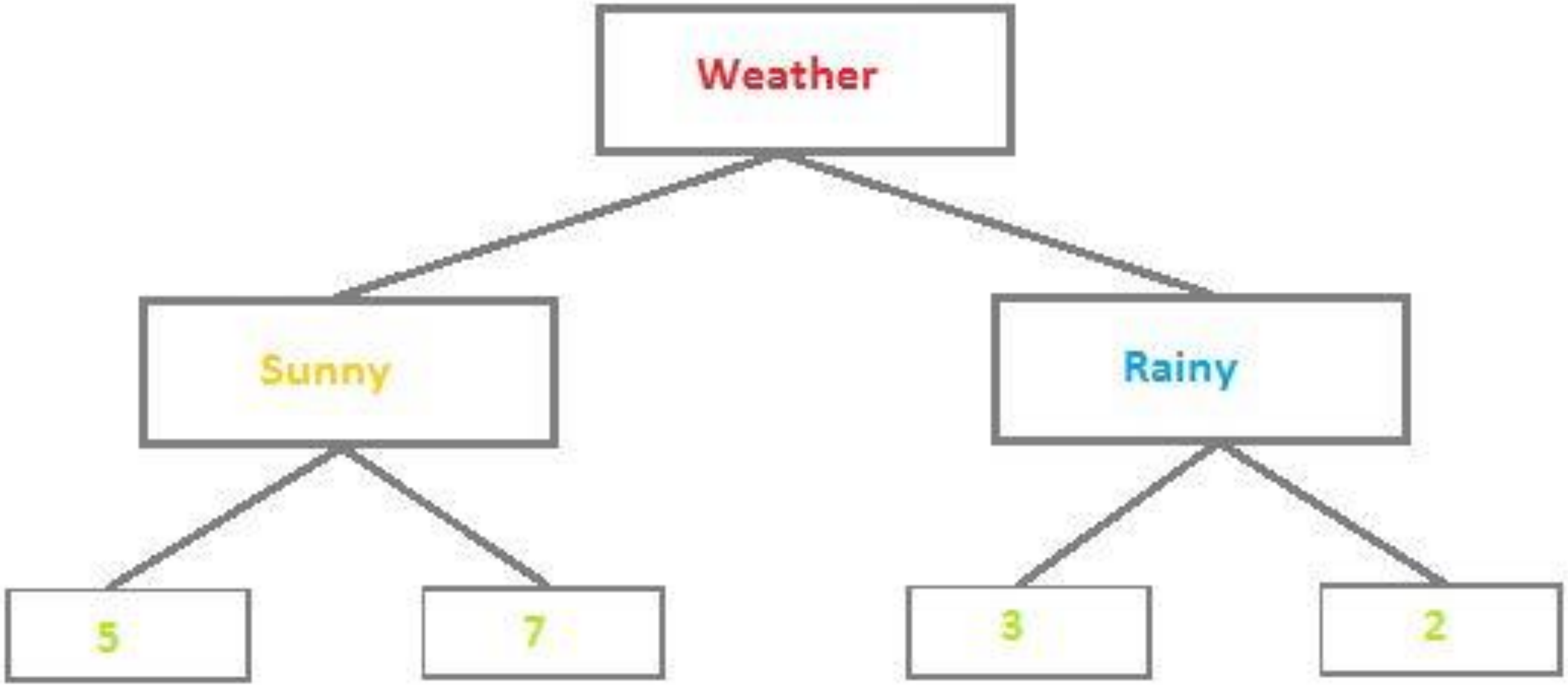
$- * - \boldsymbol{R\ Hands\ On} - * -$

Decision Trees

A decision tree is a classification machine learning method which assigns new data based on the trained model.

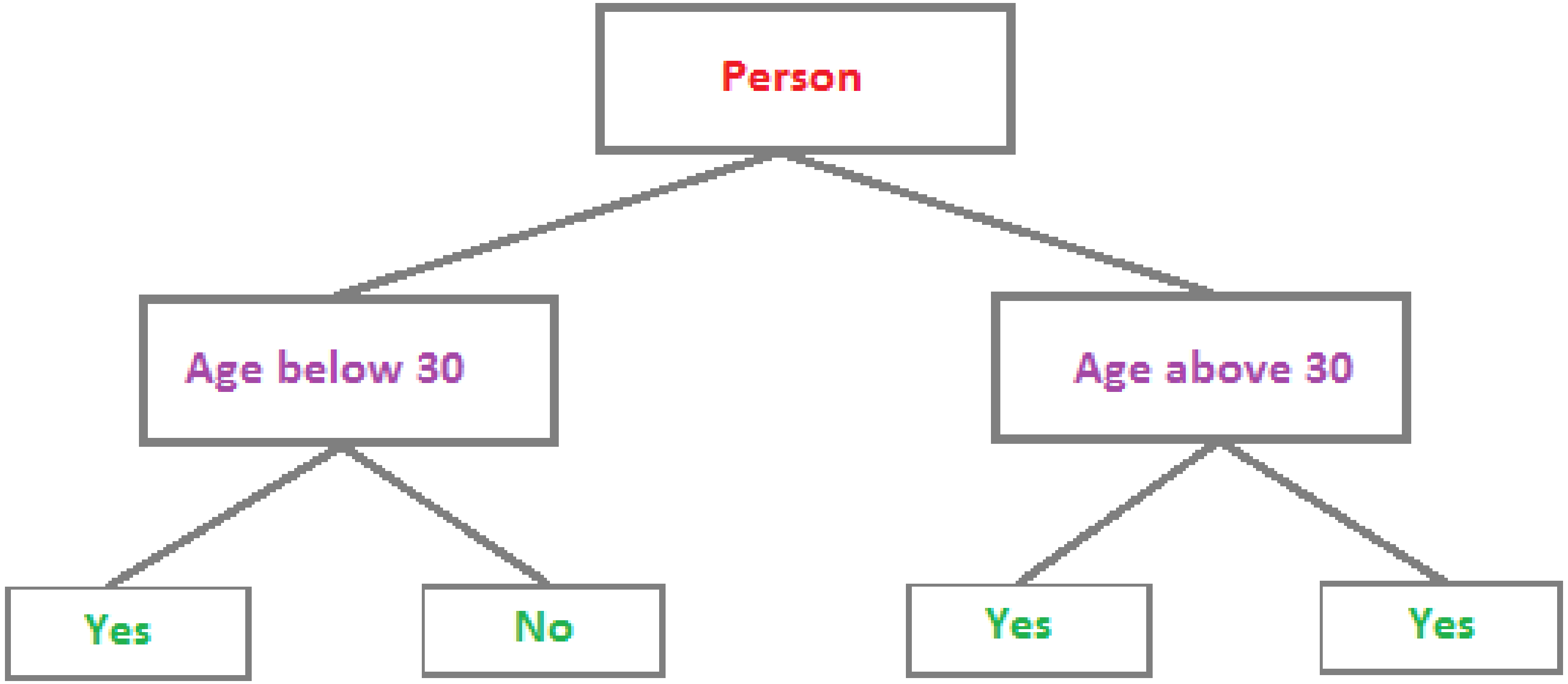
Decision trees can be implemented for both numeric (regression) as well as class categorical (classifier) output.

Regression Decision Trees



Insights: Decision tree predicting numeric response variable.

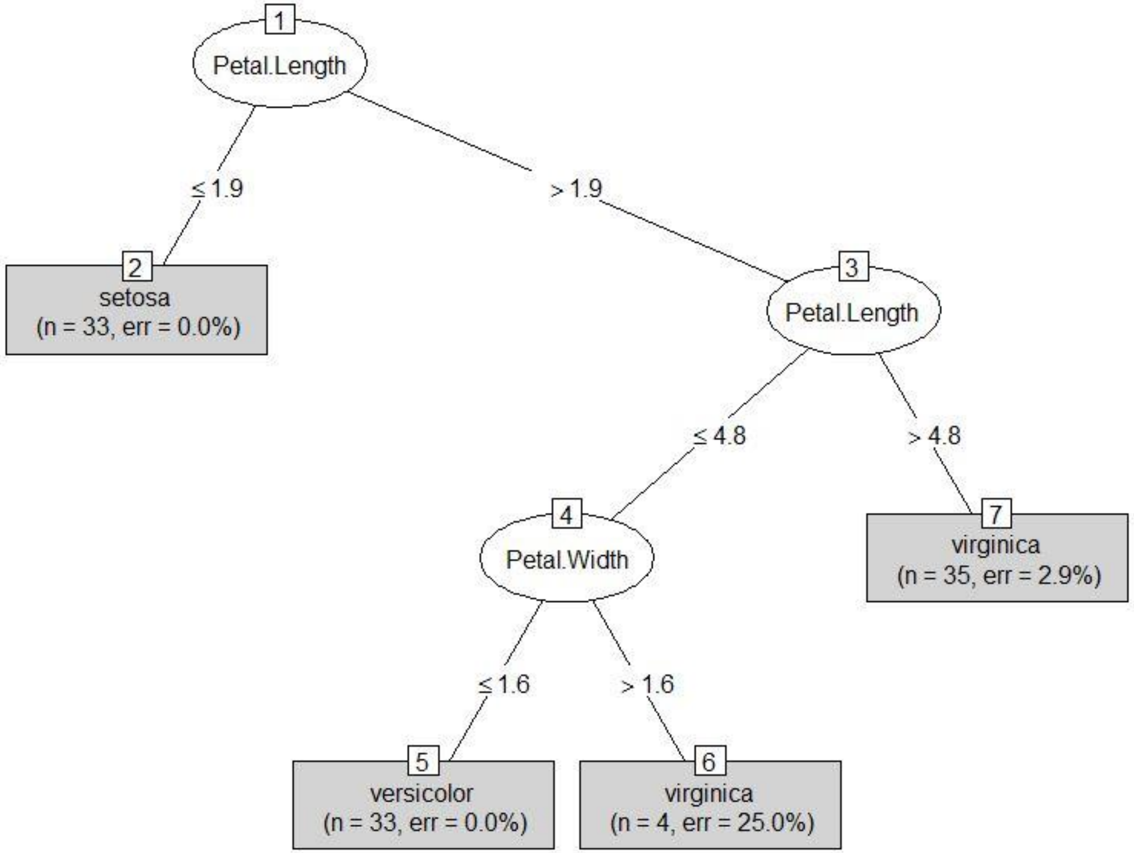
Classification Decision Trees



Insights: Decision tree predicting categorical response variable.

Decision Trees – Illustration

Consider Iris dataset where we need to predict flower class based on 4 independent variables. Let us make decision tree on iris dataset and visualize the model.



Accuracy of the model

The confusion matrix is given below for the iris dataset –

	dt_predict			
actual	setosa	versicolor	virginica	Sum
setosa	17	0	0	17
versicolor	0	12	3	15
virginica	0	0	13	13
Sum	17	12	16	45

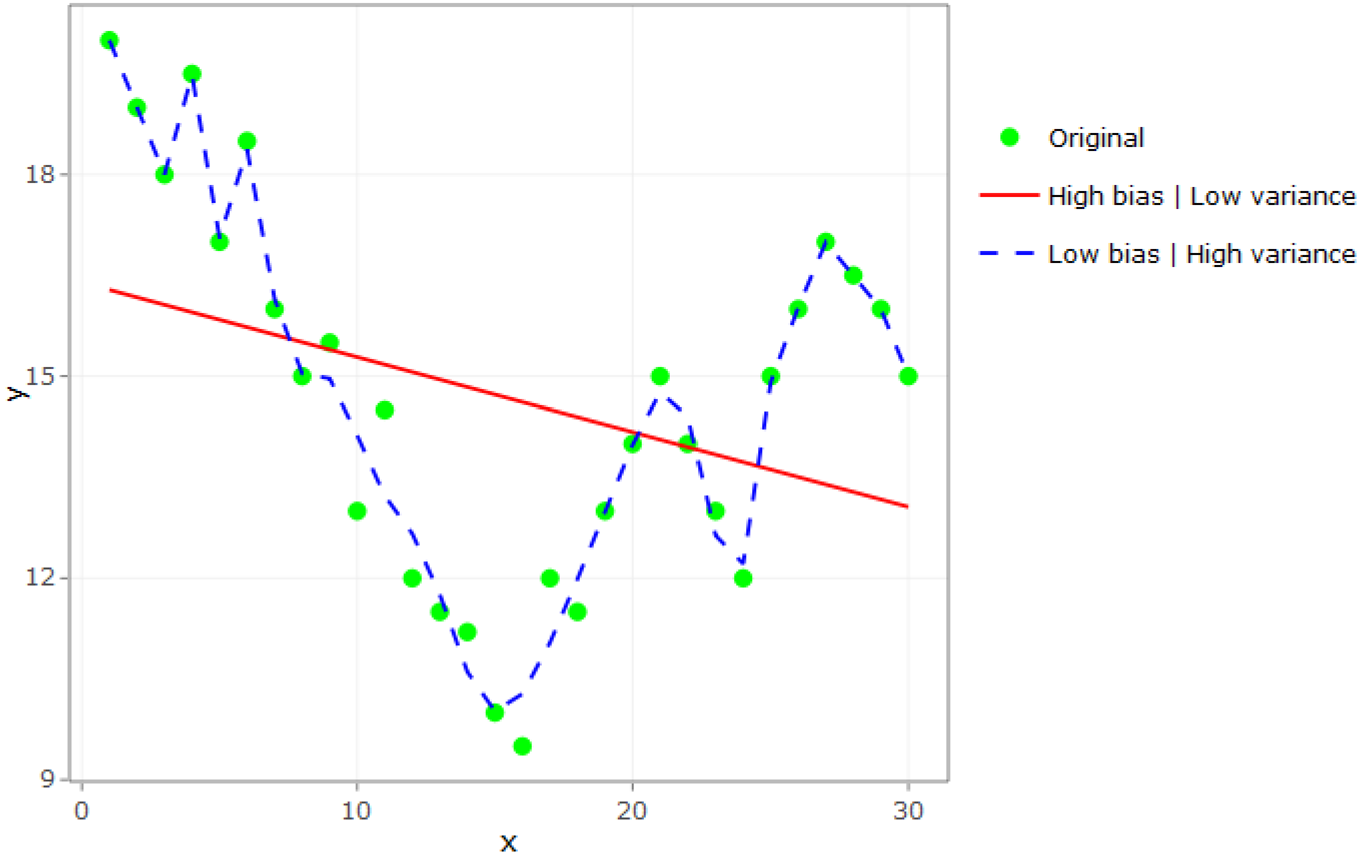
Insights: Accuracy of the given model is **93.3%**

Regularization

Regularization techniques tries to **maintain the bias – variance trade-off** in a given model by shrinking or completely removing the predictors coefficients. Consider given charts to understand how does a model looks like with different bias and variance values –

High bias | Low variance ->
Case of **under-fitting**

Low bias | High variance ->
Case of **over-fitting**



Ridge Regression

- Ridge regression adds an additional constraint to model such that the regression objective is to minimize the

$$\text{RSS} + \lambda\ (\text{L2 norm of }\beta) = \sum(\hat{y} - y)^2 + \lambda \sum \beta^2$$

- As with least squares, ridge regression tries to minimize the first factor i.e. RSS, however, the second factor called the shrinkage penalty is small when β is close to zero.
- The regularization parameter serves to control the relative impact of these two parameters. The L2 norm is applied to all the regression coefficients excluding the intercept.

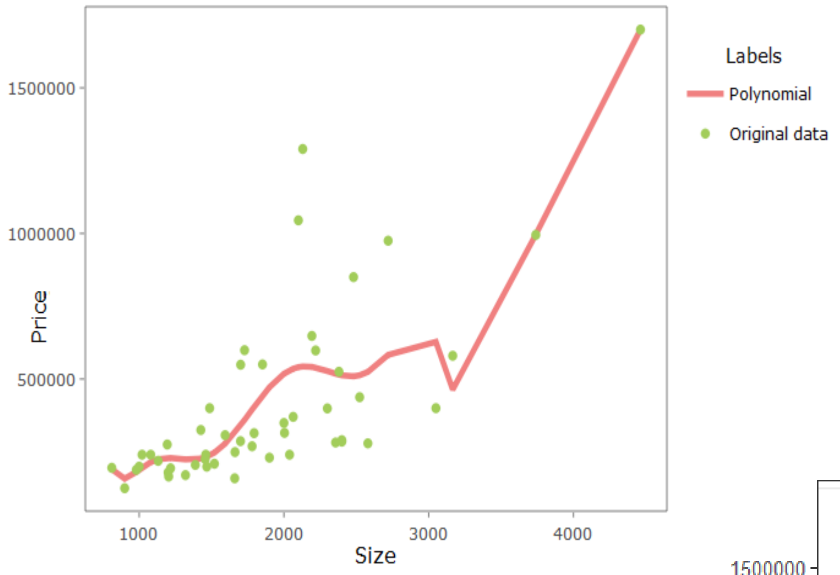
Ridge Regression – Illustration

Consider a dataset in which we need to predict the price of the house based on its size. The sample of the dataset is shown –

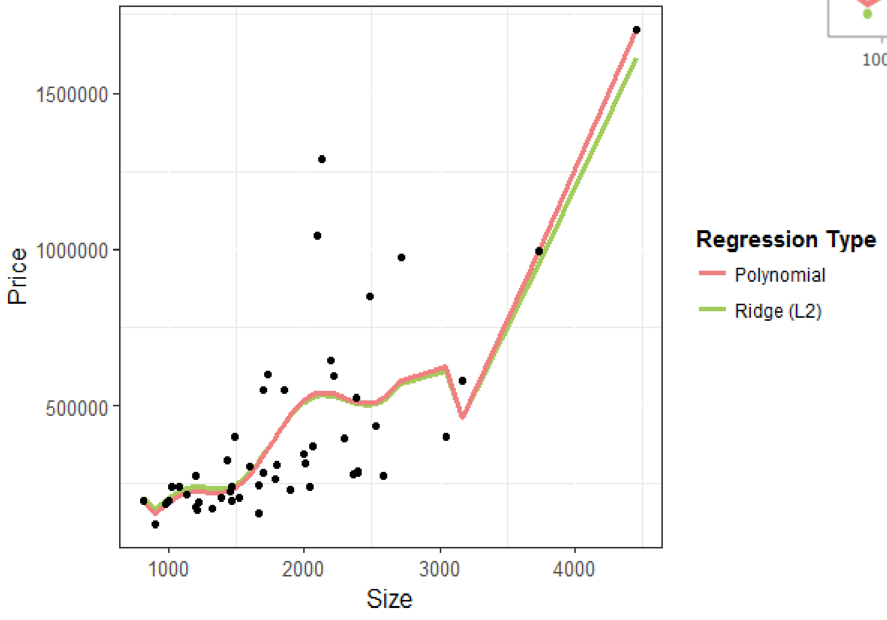
	Size	Price
1	1700	286500
2	1701	549000
3	1662	249000
4	1852	550000
5	1320	170000
6	1456	225000

Models at different range of lambda

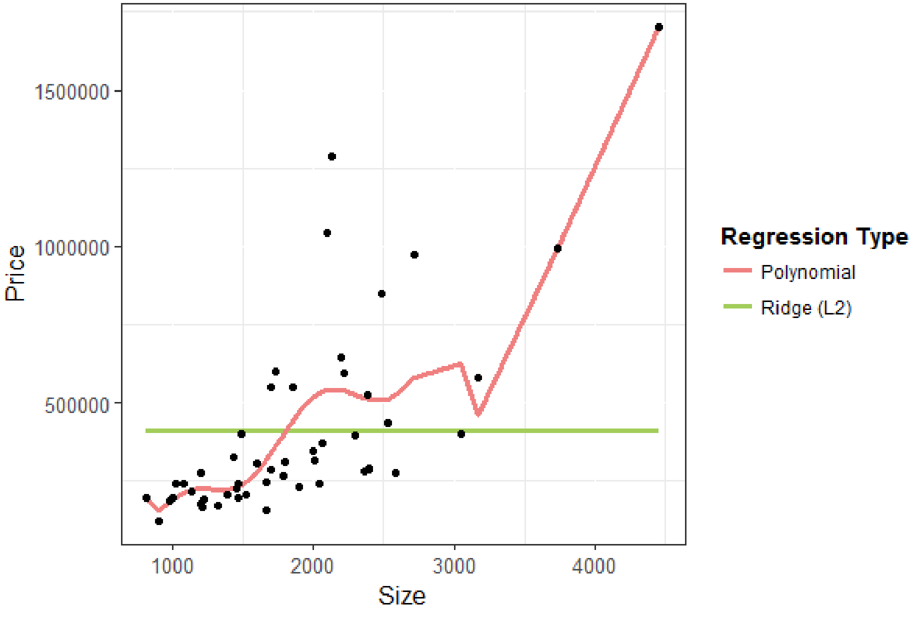
Polynomial Regression of order 10



Ridge Regression at λ = 0



Ridge Regression at λ = ∞



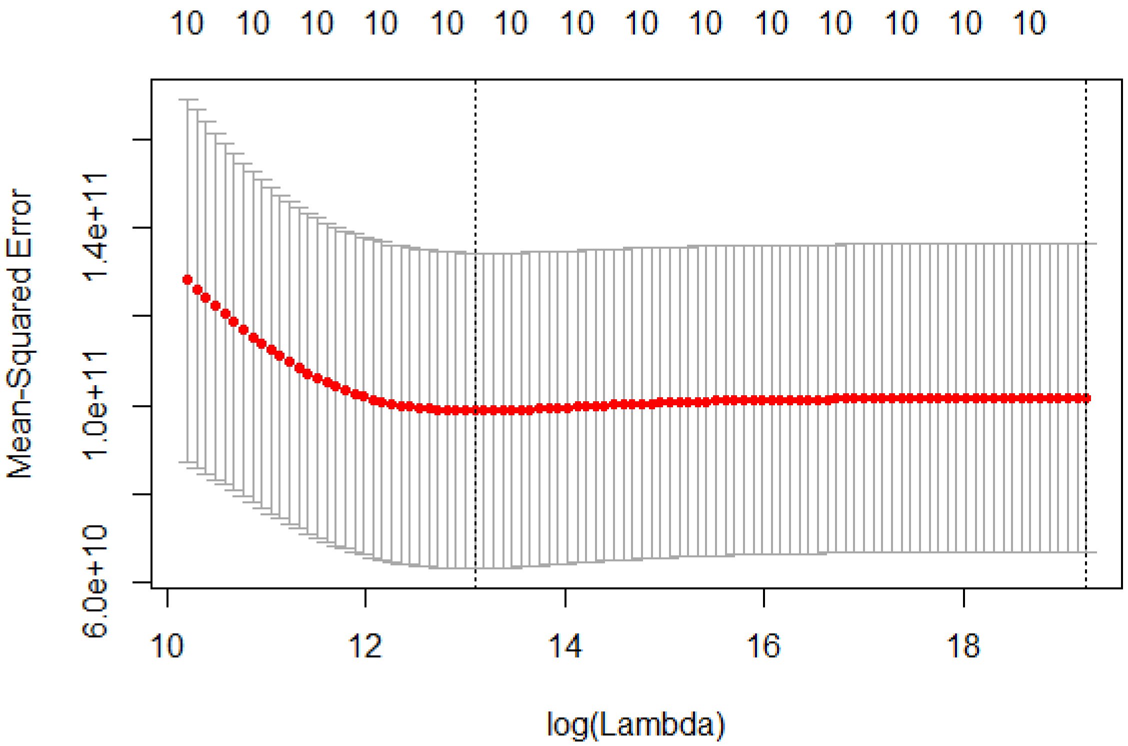
Insights:

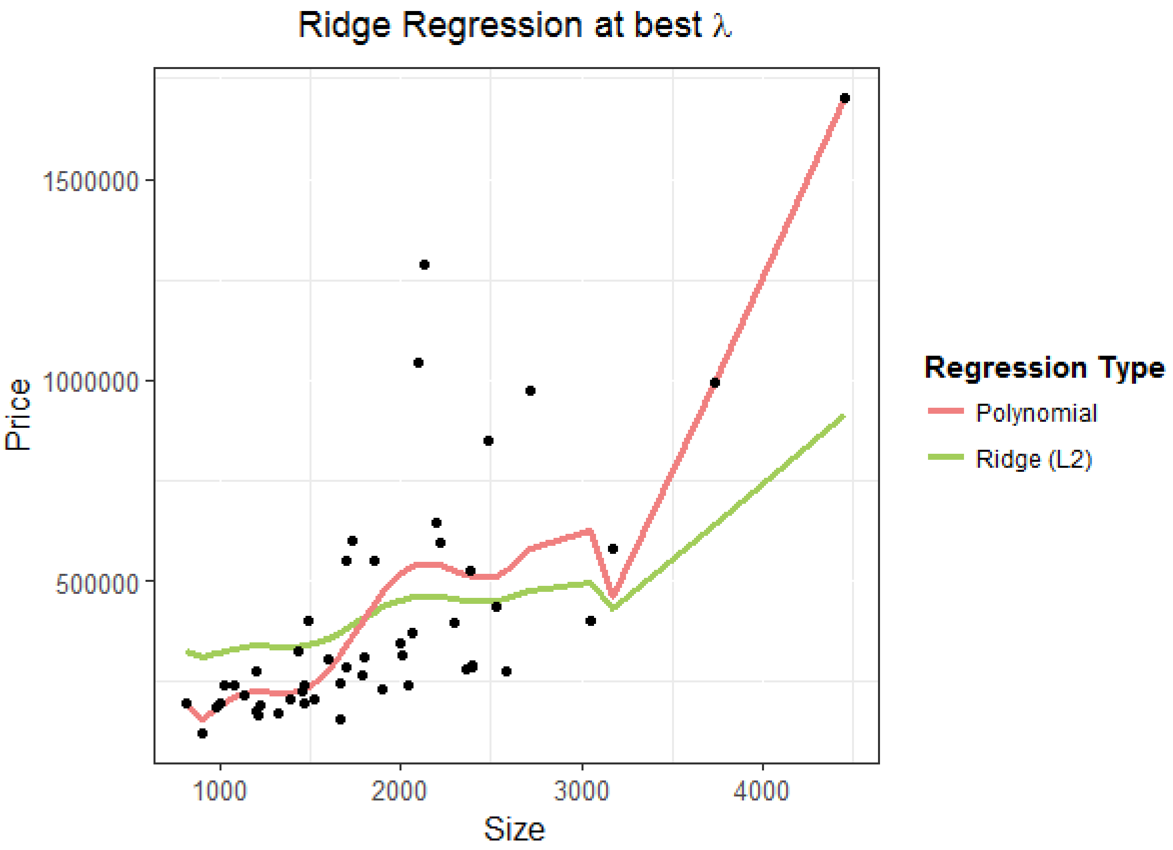
At λ = 0, regularization effect becomes 0.

At λ = infinity, RSS effect becomes 0.

Best value of lambda

The best value of lambda can be achieved using cross – validation method. The lambda with the least average CV error is chosen.





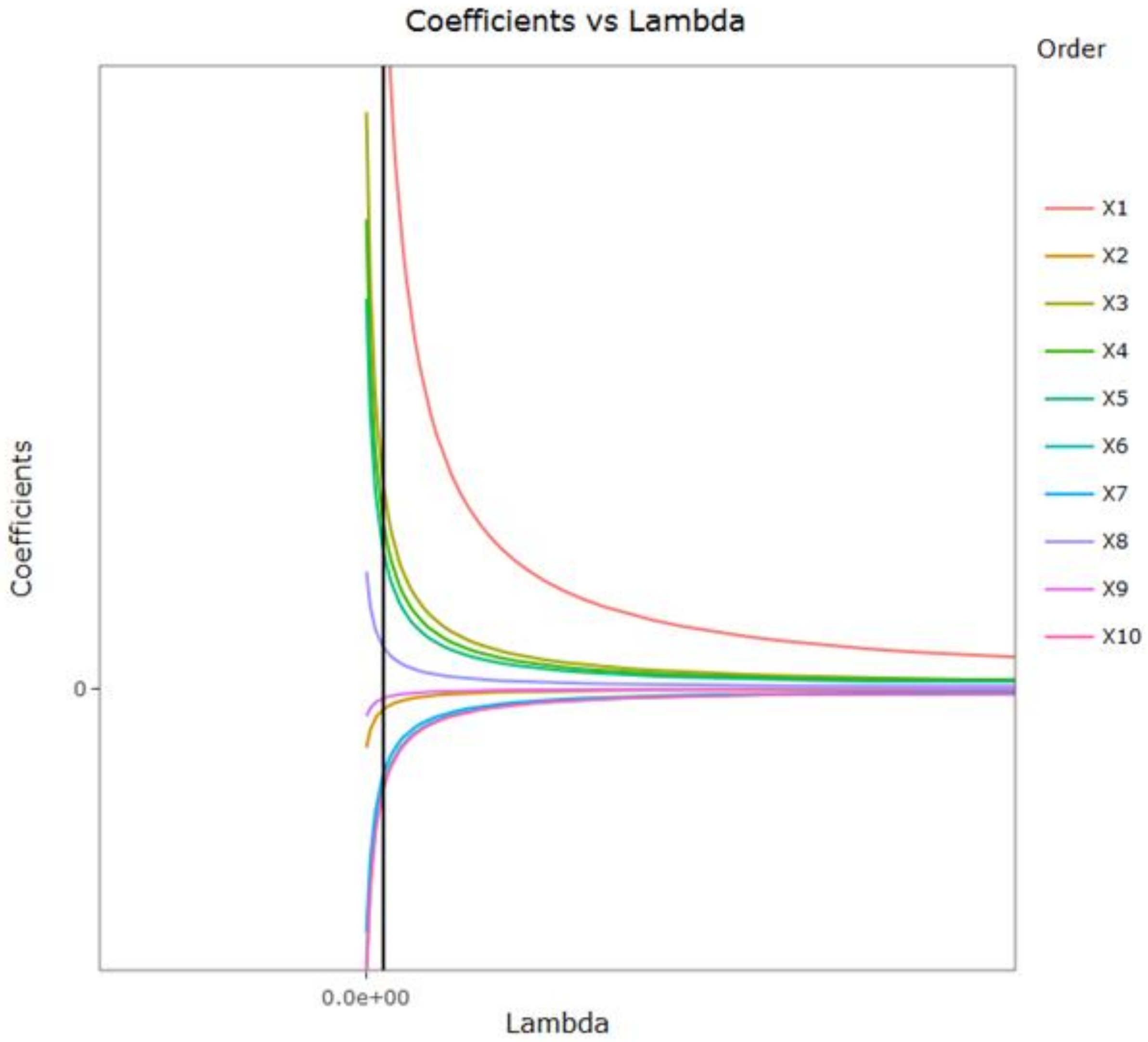
Coefficients comparison

Coefficients	Un-regularized model	Regularized model
	Order 10	Order 10
Intercept	411109	411109
β_1	1600857	699534.529
β_2	423860	185216.107
β_3	344985	150749.706
β_4	286933	125382.664
β_5	-204538	-89378.064
β_6	-179616	-78487.656
β_7	85770	37479.176
β_8	-20714	-9051.315
β_9	-211009	-92205.471
β_{10}	-43763	-19123.495

Insights:

The intercept remains intact, however regularized model coefficients have shrink.

Visualizing coefficient shrinkage with increase in lambda



Insights:

- The black line depicts the best lambda.
- Coefficient of X1 has large impact as compared with other coefficients.
- As lambda increases all the coefficients approaches zero but doesn’t become zero.

Performance comparison

RMSE	Un-regularized model	Regularized model
Train data	193216.9	238347.5
Test data	133415359	208967.4

Insights: We can infer from the given RMSE table that there is quite a remarkable improvement in the regularized model when compared to an over-fitted model.

Lasso Regression

- We observed from our last section that all of the coefficients are still intact in the model irrespective of their importance which abides to an increased complexity with little better results.

- The Lasso regularization chooses the predictor(s) along with their important coefficient(s) which has/have relatively high significant effect on prediction with increasing values of lambda.

- The lasso regularization performs L1 norm as compared with L2 regularization as shown below:

$$\sum (\hat{\boldsymbol{y}} - \boldsymbol{y})^2 + \boldsymbol{\lambda} \sum \boldsymbol{\beta}$$

- As compared with ridge regression, with the sufficient large value of λ, **lasso does forces some of the coefficient estimates to equals zero**. Hence, acting as a **variable selecto**r and reducing the complexity of the overall model.

Lasso Regression – Illustration

Consider this time having a dataset with 14 independent numeric variables and a dependent variable as listed –

"num_critic_for_reviews" "duration" "director_facebook_likes" "actor_3_facebook_likes" "actor_1_facebook_likes" "gross" "num_voted_users" "cast_total_facebook_likes" "facenumber_in_poster" "num_user_for_reviews" "budget" "actor_2_facebook_likes" "aspect_ratio" "movie_facebook_likes" "imdb_score"

Our task is to use lasso regression to select most important variables to build regression model for predicting imdb score and compare the performance with normal regression model.

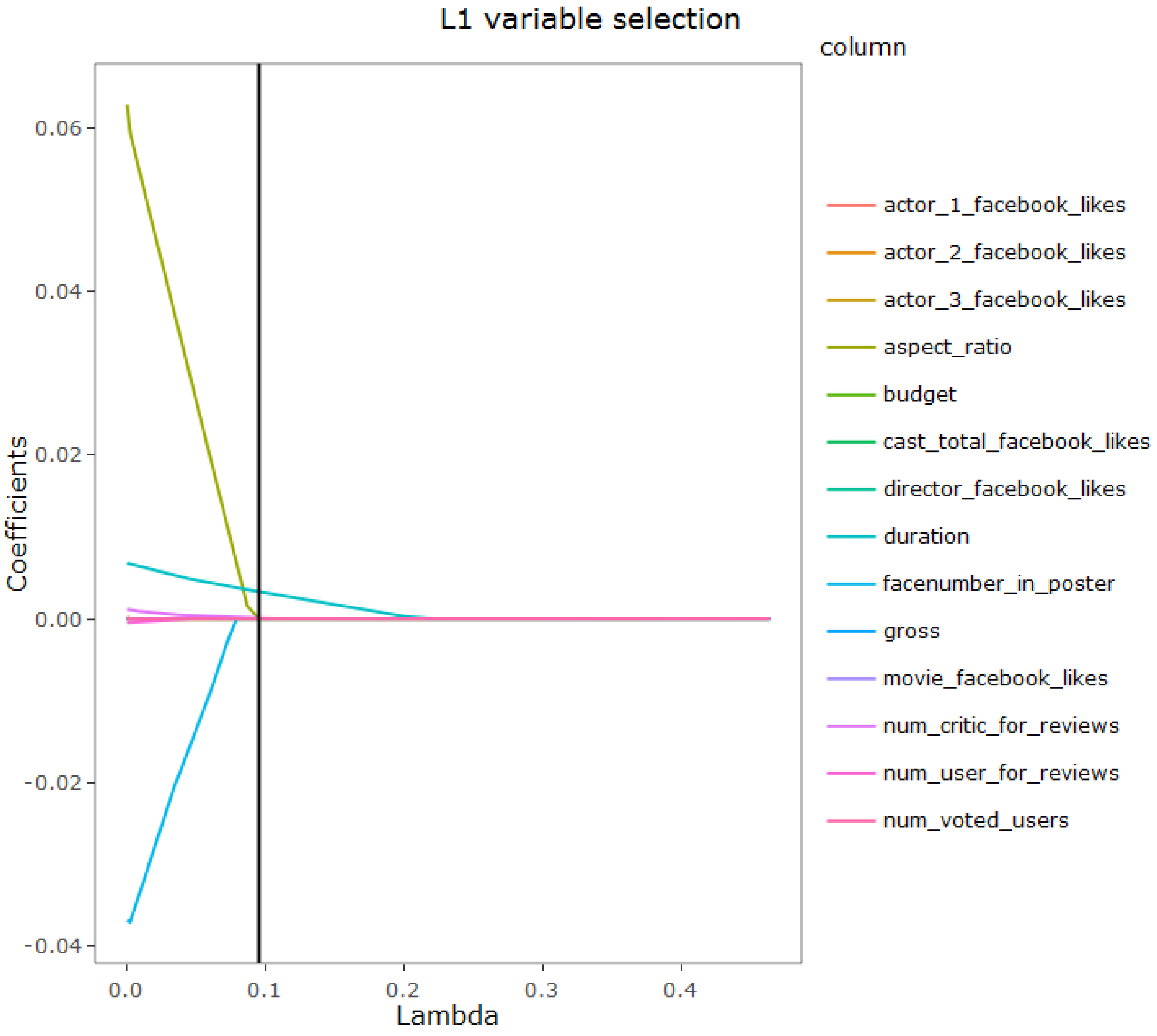
Regularized Coefficients

Once L1 model is build we can get following coefficients –

Predictors	Coefficients
(Intercept)	5.900897e+00
num_critic_for_reviews	8.062270e-05
duration	3.315895e-03
director_facebook_likes	0
actor_3_facebook_likes	0
actor_1_facebook_likes	0
gross	0
num_voted_users	2.376529e-06
cast_total_facebook_likes	0
facenumber_in_poster	0
num_user_for_reviews	0
budget	0
actor_2_facebook_likes	0
aspect_ratio	0
movie_facebook_likes	0

Insights: There are only 3 important predictors that needs to be accounted for building the model, rest all can be ignored.

Visualizing coefficient shrinkage with increase in lambda



Insights:

- The black line depicts the best lambda.
- At black line only 3 coefficients are non-zero, rest all others have become zero.
- As lambda increases all the coefficients becomes zero.

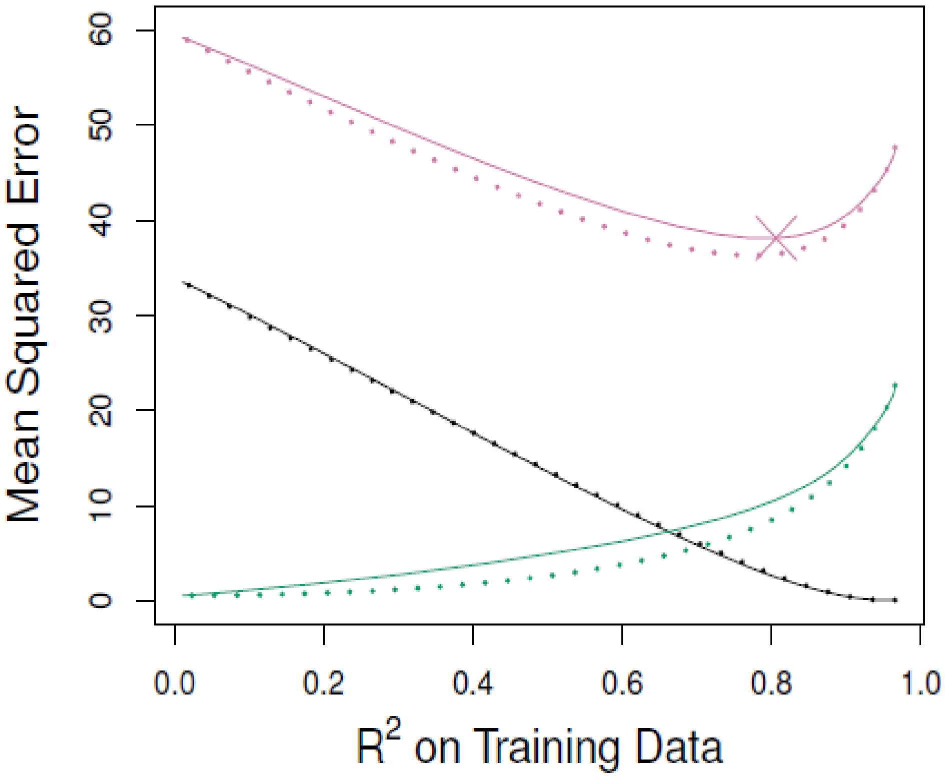
Performance comparison

RMSE	Regularized	Un-regularized
Train Data	1.00	0.96
Test Data	1.06	1.03

Insights: We can infer from the given RMSE table that though L1 regularization has similar RMSE range as that of un-regularized model but it uses only 3 predictors rather than 14. Therefore, increases model efficiency with reduced complexity.

Model Selection – Which regularization is best?

Consider first scenario in which all the given predictors are related to the response so that none can be equaled to zero.



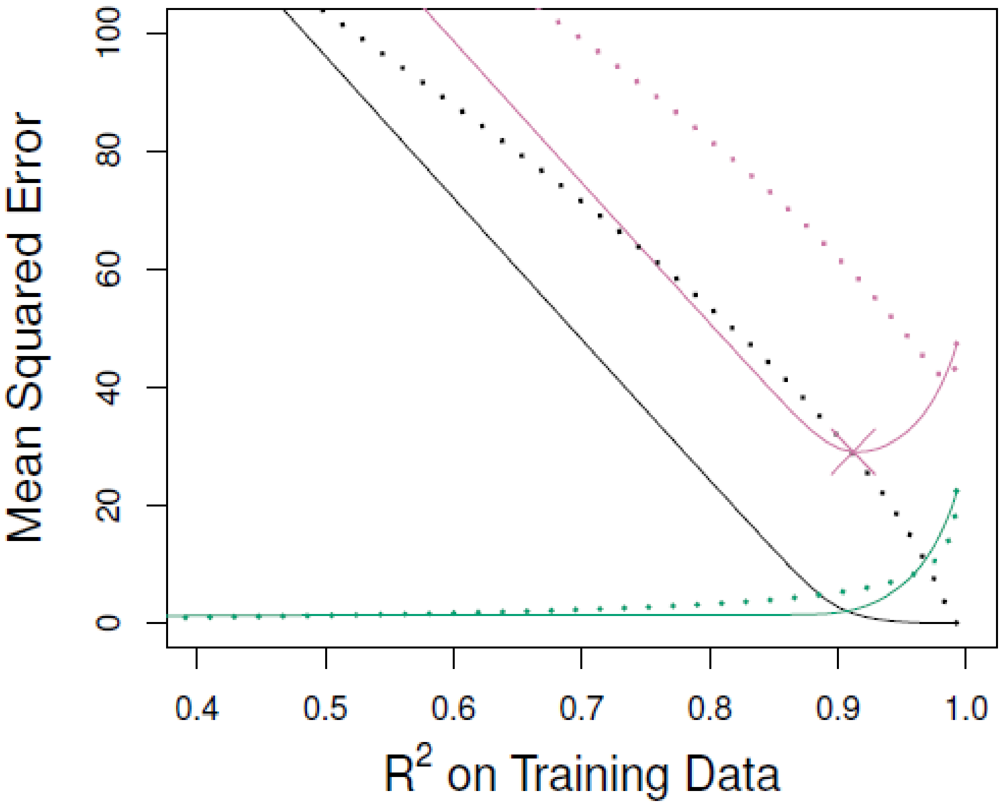
Here, green lines indicate variance; **black** lines, squared bias and red lines, test MSE.

The dotted lines resemble data for ridge regression and solid lines resemble data for lasso.

Insights: Figure reveals that variance as well as test MSE of ridge regression is lesser than that of lasso regression.

Model Selection – Which regularization is best?

Consider another scenario in which response is a function of few out of all the predictors.



Here, green lines indicate variance; **black** lines, squared bias and red lines, test MSE.

The dotted lines resemble data for ridge regression and solid lines resemble data for lasso.

Insights: Figure reveals that variance as well as test MSE of lasso regression is lesser than that of ridge regression.

Model Selection – Which regularization is best?

- This indicates that none of the method is superior to another.

- Lasso which inherently does variable selection improving interpretability performs well when few predictors are enough to decide the response

- Ridge performs well when response is a function of many variables with coefficients of roughly equal size.

- Therefore, for an un-regularized dataset, it’s better to check both the methods or a trade-off between them. This approach is well known as **elastic-net regularization**.

Elastic-Net Regularization

Elastic-Net minimizes the effect of least squares estimates along with the penalty of both ridge and lasso or minimizes the following equation –

$$\sum(\hat{\boldsymbol{y}} - \boldsymbol{y})^2 + \boldsymbol{\lambda}_2 \sum \boldsymbol{\beta}^2 + \boldsymbol{\lambda}_1 \sum \boldsymbol{\beta}$$

Elastic-Net chooses the best values of alpha and so forth lambda using **cross-validation**.

Drawback – The only drawback of Elastic-Net over Ridge and Lasso is that it is **computationally expensive** as it builds model for all the range of alpha [0, 1] where, alpha = 0 (Ridge) and alpha = 1 (Lasso).

Ensemble methods

- Ensemble learning models improves the performance of a given model by combining several results together whose performance stands out when compared with a single model result.

- Ensemble methods types –
  1. Bootstrap Aggregation (Bagging)
  2. RandomForests

- To understand the functioning of each of these types we need to understand the concept of **bootstrap**.

Bootstrap

Bootstrap helps in obtaining new distinct sample sets by repeatedly sampling observations from the original data set.
The sampling is performed *with replacement* i.e. similar observation values can occur in the same sample set.

Let us take a dummy dataset –

X	Y
2.3	4.5
5.6	9.6
3.6	8.0

Bootstrap works to minimize the term –

$$Var(\propto X + \ (1 - \propto)Y)$$

where, the parameter $\propto$ is given by –

$$\propto = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}}$$

Bootstrap

Let us create 3 different bootstrap datasets to be marked as B. For each of the B datasets, we'll get corresponding ∝ values.

Table given below depicts the B bootstrap datasets along with their ∝.

X	**Y**
2.3	4.5
5.6	9.6
3.6	8.0

	Original		$B = 1$		$B = 2$		$B = 3$	
	X	Y	X	Y	X	Y	X	Y
Observation 1	2.3	4.5	2.3	4.5	2.3	4.5	5.6	9.6
Observation 2	5.6	9.6	3.6	8.0	5.6	9.6	3.6	8.0
Observation 3	3.6	8.0	3.6	8.0	5.6	9.6	3.6	8.0
∝	1.97		1.59		2.83		-4.0	

We can now find standard error estimate (here 3.63) between bootstrapped ∝ values and original ∝ value using the equation given below –

$$SE_B(\alpha_{orig}) = \sqrt{\frac{1}{B-1}\sum_{i=1}^{B}\left(\alpha_i - \frac{1}{B}\sum_{j=1}^{B}\alpha_i\right)^2}$$

Insights: We obtained a small (3.63) standard error; hence we can use bootstrap samples to form different models and later perform further analysis on the results.

Bootstrap Aggregation (Bagging)

- As we have observed through bootstrap that we can create distinct data samples from a same dataset with a variance approximately equal to when compared with model if built with individual dataset.
- The fundamental concept of bagging is to –
  1. Take many training sets from the population
  2. Build a separate prediction model using each training set
  3. Average the resulting predictions.

- In terms of tree, governing on the principle of bootstrap, we make B separate training sets with functions $f_1(x)$, $f_2(x)$ …. . . $f_B(x)$ and average them in order to obtain a single model with least variance given by –

$$f_{bag}(x) = \frac{1}{B}\sum_{i=1}^{B} f_i(x)$$

- Bagging can be applied to both regression and classification models.
- A bagged decision tree is not pruned and forms quite deep. Therefore, **each individual tree has high variance and low bias**.
- Averaging regression trees in the end reduces variance.
- In case of classification trees, we can mark the category predicted by each B trees and finally the majority vote wins.

Bootstrap Aggregation (Bagging) – Illustration

- Consider predicting house price based on 4 independent variables namely Location, Bedroom, Bathroom and Size.
- Split the data into training and testing parts.
- From the training data, split it into 2 halves of equal size and form decision trees on both the data.
- Calculate train and test RMSE for both halves.

RMSE	Tree 1	Tree 2
Train Data	146846.4	217557.1
Test Data	219506.9	591291.5

Insights: As can be inferred from the table, the RMSE of both trees are quite different yet both trees are arrived from same training data.

This problem occurs due to ***high variance*** in given trees. To reduce such high variance, we use bagging.

Performance comparison

	RMSE	Simple tree	Bagged tree
Tree 1	Train Data	146846.4	209371.5
	Test Data	219506.9	173166.7
Tree 2	Train Data	217557.1	311188.8
	Test Data	591291.5	199828.8

Tree 1 + Tree 2	Minimum, Maximum	Range
Simple Tree	146846.4, 591291.5	444445.1
Bagged Tree	173166.7, 311188.8	138022.1

Insights:

- RMSE values of bagged tree for training and testing data are quite closer for both trees as compared with simple tree.
- Range of RMSE values of bagged tree is quite less than as compared to simple tree.

Interpretability and Variable Importance

- Building bagged trees **losses the interpretability of the model**.
- Significant predictor can be retrieved using *importance* variable in R.
- For instance, important variable for bagged tree 1 can be listed in sequence of their mean decrease in accuracy

	%IncMSE	IncNodePurity
Location	1.95E+09	1.03E+12
Bedrooms	-6366563	6.66E+09
Bathrooms	2.41E+08	1.71E+10
Size	8.72E+09	1.40E+11

Insights:

- A higher value of %IncMSE is considered to be important variable. Complementary to this is IncNodePurity, again whose larger value is preferred.
- Range of RMSE values of bagged tree is quite less than as compared to simple tree.

– * – R Hands On – * –

RandomForests

- Bagging too faces an issue of partiality. Recall, that we’ve **passed all the predictors to the bagged model**,

- So in case if there happens to be a **dominating predictor** then for each tree made its effect will overcome the effect of other relevant predictors.

- Therefore, there’s a need to remove this static predictor which is done usually by **passing only square root number of predictors** to the model.

- This will ensure that **for each new run different predictors are chosen** and hence suppressing the dominating effect of one or two predictors. This phenomenon is also coined as *random forests.*

RandomForests – Illustration

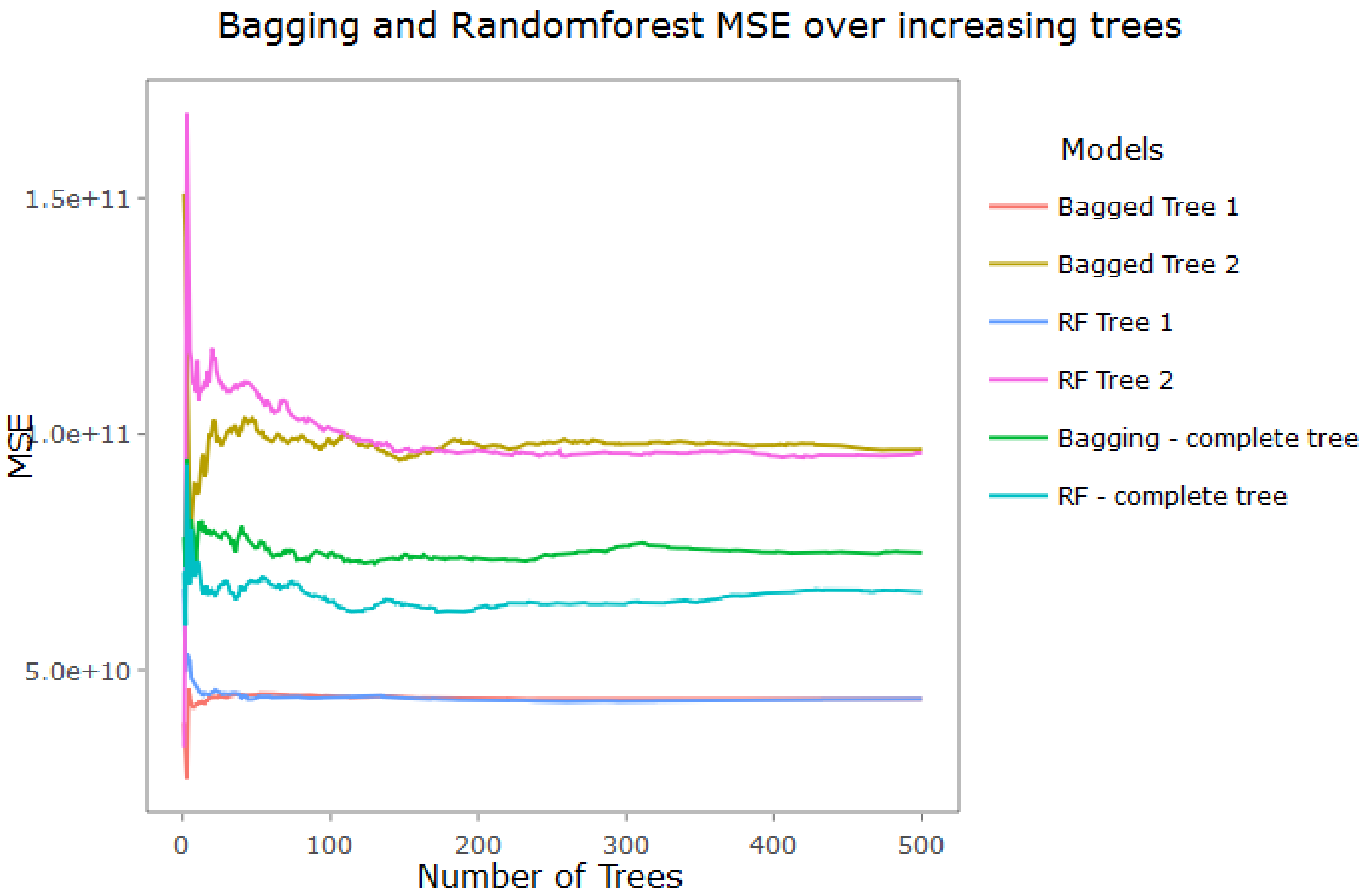
We perform implementation of random forest on the same data that we used for bagging.

Model building is similar to bagging except this time we need to pass 2 predictors (square root of 4).

	RMSE	Simple tree	Bagged tree	Random Forest
Tree 1	Train Data	146846.4	209371.5	209653.3
	Test Data	219506.9	173166.7	119622.6
Tree 2	Train Data	217557.1	311188.8	309596.7
	Test Data	591291.5	199828.8	158575.1

Insights: For the given data Random Forest lacks when compared with the performance of bagging. However, as the number of trees increases the performance of random forests overcome the performance of bagging.

Bagging vs RandomForests Performance with increasing trees



Insights:

With rise in the number of trees –

- Random forest improves on Tree 1 and Tree 2 when compared with bagging.
- On overall training data, Random forest clearly advances bagging with less MSE.

*– * – R Hands On – * –*

Boosting

- Boosting is one of the important technique in ML which helps to improve the prediction of a model.
- It can be applied both on classification as well as on regression problems.

- The **fundamental principle** behind all boosting algorithm is that they improve model accuracy by *boosting weak learners to become strong learners*. The way they convert a weak learner into a strong learner has given rise to different algorithms like –
  1. Adaptive Boosting
  2. Gradient Boosting
  3. Extra Gradient Boosting

- Here, we will discuss the first and easiest among above three namely, adaptive boosting.

Adaptive Boosting – Algorithm

Adaptive Boosting is adaptive in the sense that it keeps on updating its weight corresponding to each data points assigning higher weights to weak learners and lower weights to strong learners.

The algorithm for AdaBoost is given below –

1. Initialize equal weights to all training points

$$W = \frac{1}{N}$$

2. Calculate error rate for each classifier

$$\varepsilon = \sum W_{\mathrm{i}}$$

3. Pick best classifier with smallest error rate

4. Calculate voting power for classifier

$$\propto = \frac{1}{2} ln\left(\frac{1-\varepsilon}{\varepsilon}\right)$$

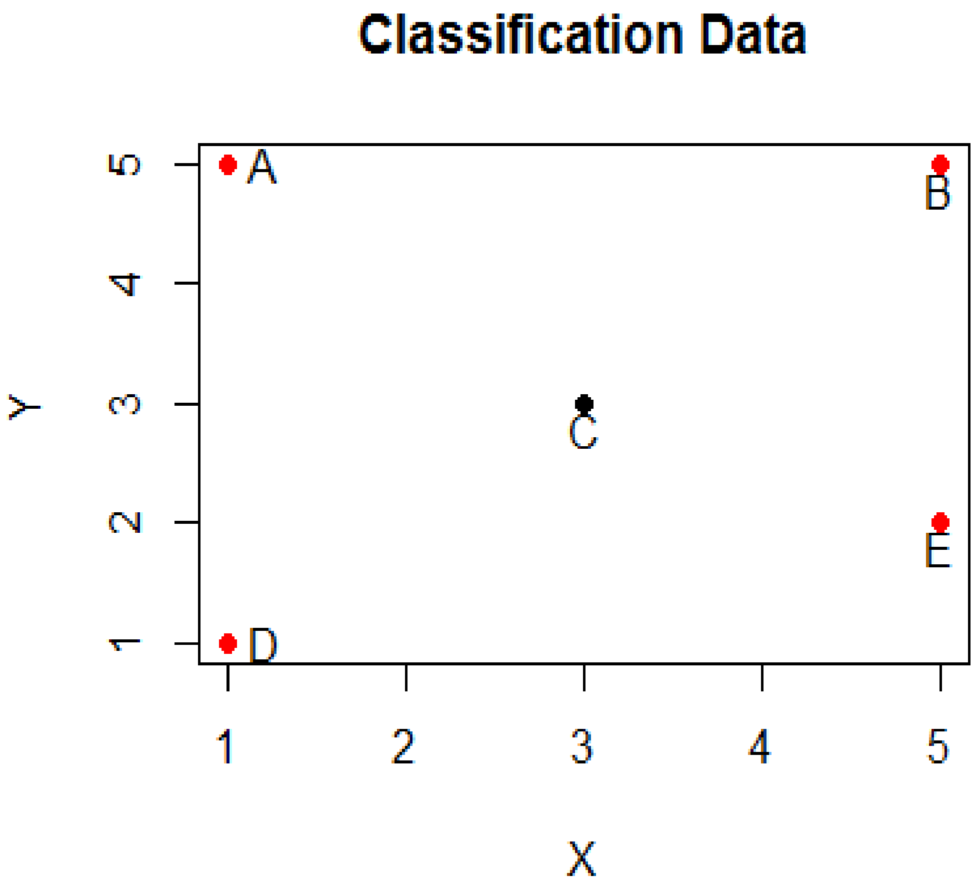
5. Reached end? If no, update weights to emphasize points that were misclassified as shown below and go back to step 2 for recalculating error rate.

$$W_{new} = \begin{cases} \frac{1}{2}\frac{1}{1-\varepsilon}W_{old} & if, right \\ \frac{1}{2}\frac{1}{\varepsilon}W_{old} & if, wrong \end{cases}$$

Adaptive Boosting – Illustration

Consider a dummy dataset as shown. X and Y are independent variables are T is the dependent variable. The complete dataset along with its visualization is given below –

X	Y	T
3	3	No
1	1	Yes
5	2	Yes
1	5	Yes
5	5	Yes



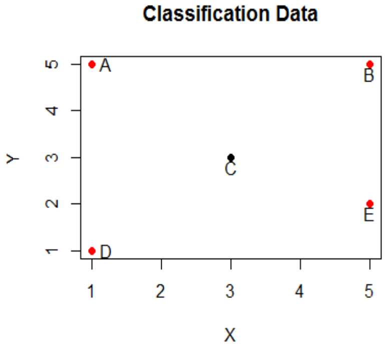
Adaptive Boosting – Steps

Step 1 –

Initialize equal weights to all training points. Since we’ve 5 data points, therefore each data point get assigned with an equal weight of 0.2.

Weights	Iteration 1
W_A	0.2
W_B	0.2
W_C	0.2
W_D	0.2
W_E	0.2

Adaptive Boosting – Steps



Step 2 –

Defining the classifier boundaries and misclassified points corresponding to them:

Iteration	Classifier boundary	Misclassified points	Error rate (ε)	Comment
1	X>=2	A C D	0.6	Considering all the points greater than or equal to 2 in the category of 1 and 0 for the rest. A and D: are misclassified as 0 C: is misclassified as 1
	X<2	B E	0.4	Considering all the points less than 2 in the category of 1 and 0 for the rest. B and E: are misclassified as 0

Adaptive Boosting – Steps

Step 3 –

Next up, let us pick the classifier with least error rate, here X<2 with $\boldsymbol{\varepsilon}$ **=** $\frac{2}{5}$ or 0.4

Step 4 –

To calculate the voting power as shown below: $\propto = \frac{1}{2} ln\left(\frac{1-\varepsilon}{\varepsilon}\right)$

Since $\boldsymbol{\varepsilon} = \mathbf{0.4}$, therefore $\propto = 0.2027$

Adaptive Boosting – Steps

Step 5 –

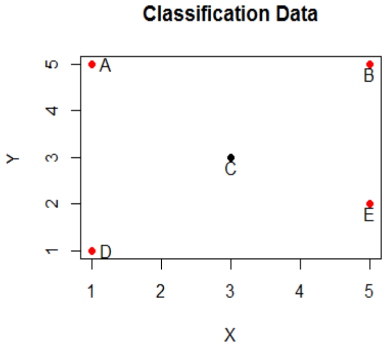
Moving further with second iteration and hence reassigning new weights by formula:

$$W_{new} = \begin{cases} \frac{1}{2}\frac{1}{1-\varepsilon}W_{old} & if, right \\ \frac{1}{2}\frac{1}{\varepsilon}W_{old} & if, wrong \end{cases}$$

With X<2 only 2 training points i.e. B and E aren’t classified in their right category rest 3 points i.e. A, C and D are classified correctly.

Weights	Iteration 1	Iteration 2
W_A	0.2	0.166
W_B	0.2	0.250
W_C	0.2	0.166
W_D	0.2	0.166
W_E	0.2	0.250

Adaptive Boosting – Steps



Recalculating error rate –

Iteration	Classifier boundary	Misclassified points	Error rate (ε)	Comment
2	X<4	B C E	0.666	Considering all the points less than 2 in the category of 1 and 0 for the rest. B and E: are misclassified as 0 C: is misclassified as 1
	X>=4	0	0	Considering all the points greater than or equal to 4 in the category of 1 and 0 for the rest. No misclassified points

Insights: Points A, C and D were classified right in iteration 1 for X < 2. Using X>=4 we also classified points B and E rightly.

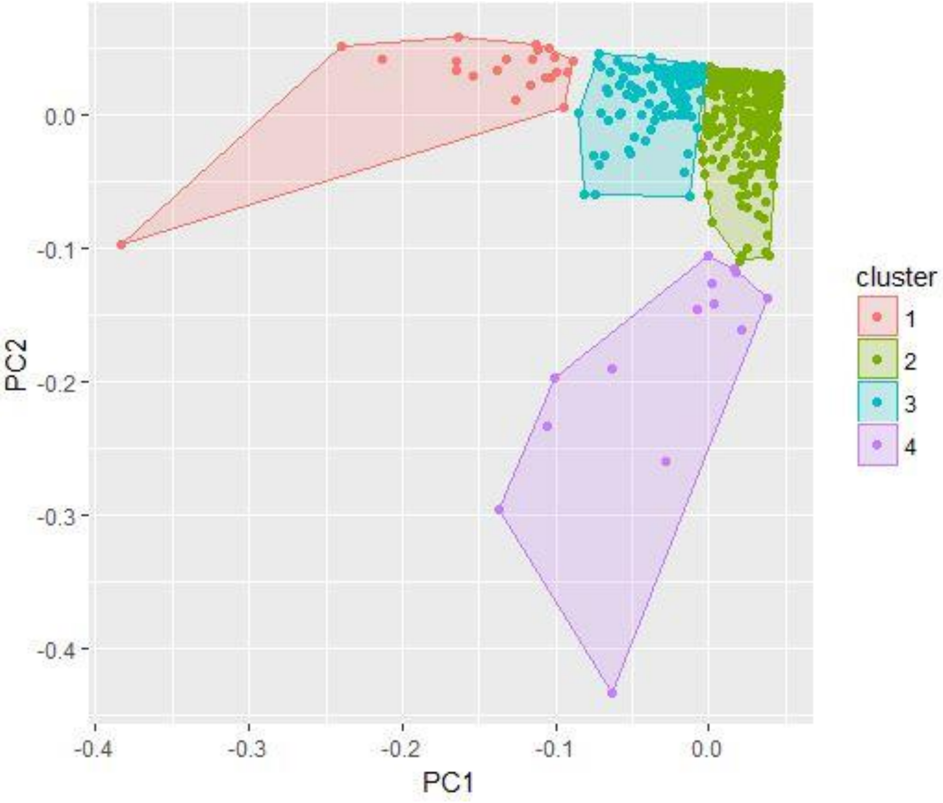
– * – ***R Hands On*** – * –

Clustering

Clustering is a type of unsupervised ML technique which forms clusters based on the density of data in the plot.
The commonly used clustering algorithm is k-means.

Steps in k-means:

- Pass value of k; the number of clusters.
- Each centroid defines one of the clusters.
- Each data point is assigned to its nearest centroid
- Centroids are recomputed by accounting for the mean of all the data points assigned to that cluster



$-\ * - R\ Hands\ On - * -$