Unit III: Regression Analysis

Linear multiple regression, Estimation and testing of coefficients, R^2 and adjusted R^2 coefficients Logistic regression, Estimation and Testing of coefficients, K – Nearest Neighbor classifier, random forest, classifycation errors, Ridge Regression and Support Vector Machine.

Linear Regression:

Straight-line regression analysis involves a response variable, y, and a single predictor variable, x. It is the simplest form of regression, and models y as a linear function of x.

That is,

$$y = w_0 + w_1 x.$$

where the variance of y is assumed to be constant, and w_0 and w_1 are regression coefficients specifying the Y-intercept and slope of the line, respectively.

These coefficients can be solved for by the method of least squares, which estimates the best-fitting straight line which minimizes the error between the actual data and the estimate of the line. Let D be a training set consisting of values of predictor variable, x, for some population and their associated values for response variable, y. The training set contains |D| data points of the form (x1, y1), (x2, y2),..., (x|D| , y|D|).12 The regression coefficients can be estimated using this method with the following equations:

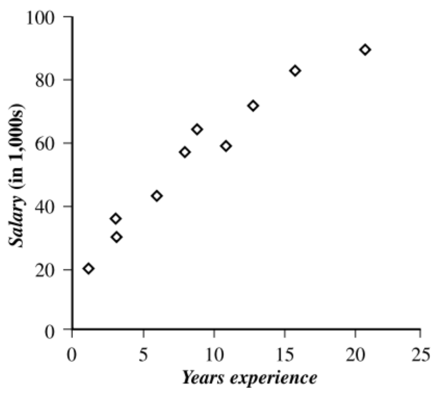
$$w_1 = \frac{\sum_{i=1}^{|D|}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{|D|}(x_i - \bar{x})^2}$$

$$w_0 = \bar{y} - w_1 \bar{x}$$

where $\bar{x}$ is the mean value of $x_1, x_2, \ldots, x_{|D|}$, and $\bar{y}$ is the mean value of $y_1, y_2, \ldots, y_{|D|}$.

Example:

x years experience	*y salary (in \$1000s)*
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83



We model the relationship that salary may be related to the number of years of work experience with the equation $y = w_0 + w_1 x$.

Given the above data, we compute $\bar{x} = 9.1$ and $\bar{y} = 55.4$. Substituting these values into Equations (6.50) and (6.51), we get

$$w_1 = \frac{(3-9.1)(30-55.4)+(8-9.1)(57-55.4)+\cdots+(16-9.1)(83-55.4)}{(3-9.1)^2+(8-9.1)^2+\cdots+(16-9.1)^2} = 3.5$$

$$w_0 = 55.4 - (3.5)(9.1) = 23.6$$

Thus, the equation of the least squares line is estimated by $y = 23.6 + 3.5x$.
we can predict that the salary of a college graduate with, say, 10 years of experience is
=23.6+3.5*10
=58,600.

Multiple linear regression: **https://www.statology.org/multiple-linear-regression/**
https://www.statology.org/multiple-linear-regression-by-hand/
http://faculty.cas.usf.edu/mbrannick/regression/Reg2IV.html

It is an extension of straight-line regression which involve more than one predictor variable. It allows response variable y to be modeled as a linear function of, say, n predictor variables or attributes, A1, A2,..., An, describing a tuple, X. (That is, X = (x1, x2,..., xn).)
Our training data set, D, contains data of the form (X1, y1), (X2, y2), ... , (X|D| , y|D|), where the Xi are the n-dimensional training tuples with associated class labels, yi .
An example of a multiple linear regression model based on two predictor attributes or variables, A1 and A2, is

$$y = w_0 + w_1 x_1 + w_2 x_2,$$

where x1 and x2 are the values of attributes A1 and A2, respectively, in X. The method of least squares is used to solve for w0, w1, and w2.
For two variable case,

$$w_1 = \frac{(\sum x_2^2)(\sum x_1 y) - (\sum x_1 x_2)(\sum x_2 y)}{(\sum x_1^2)(\sum x_2^2) - (\sum x_1 x_2)^2}$$

$$w_2 = \frac{(\sum x_1^2)(\sum x_2 y) - (\sum x_1 x_2)(\sum x_1 y)}{(\sum x_1^2)(\sum x_2^2) - (\sum x_1 x_2)^2}$$

and

$$a = \bar{Y} - b_1 \bar{X}_1 - b_2 \bar{X}_2$$

where a= w_0 , B_1=w_1 and b_2= w_2

Example: following dataset with one response variable *y* and two predictor variables X_1 and X_2:

y	X_1	X_2
140	60	22
155	62	25
159	67	24
179	70	20
192	71	15
200	72	14
212	75	14
215	78	11

Step 1: Calculate X_1^2, X_2^2, X_1y, X_2y and X_1X_2.

	y	X_1	X_2
	140	60	22
	155	62	25
	159	67	24
	179	70	20
	192	71	15
	200	72	14
	212	75	14
	215	78	11
Mean	181.5	69.375	18.125
Sum	1452	555	145

	X_1^2	X_2^2	X_1y	X_2y	X_1X_2
	3600	484	8400	3080	1320
	3844	625	9610	3875	1550
	4489	576	10653	3816	1608
	4900	400	12530	3580	1400
	5041	225	13632	2880	1065
	5184	196	14400	2800	1008
	5625	196	15900	2968	1050
	6084	121	16770	2365	858
Sum	38767	2823	101895	25364	9859

Step 2: Calculate Regression Sums.

Next, make the following regression sum calculations:

- $\Sigma x_1^2 = \Sigma X_1^2 - (\Sigma X_1)^2 / n = 38{,}767 - (555)^2 / 8 = \mathbf{263.875}$
- $\Sigma x_2^2 = \Sigma X_2^2 - (\Sigma X_2)^2 / n = 2{,}823 - (145)^2 / 8 = \mathbf{194.875}$
- $\Sigma x_1 y = \Sigma X_1 y - (\Sigma X_1 \Sigma y) / n = 101{,}895 - (555{*}1{,}452) / 8 = \mathbf{1{,}162.5}$
- $\Sigma x_2 y = \Sigma X_2 y - (\Sigma X_2 \Sigma y) / n = 25{,}364 - (145{*}1{,}452) / 8 = \mathbf{-953.5}$
- $\Sigma x_1 x_2 = \Sigma X_1 X_2 - (\Sigma X_1 \Sigma X_2) / n = 9{,}859 - (555{*}145) / 8 = \mathbf{-200.375}$

	y	X_1	X_2
	140	60	22
	155	62	25
	159	67	24
	179	70	20
	192	71	15
	200	72	14
	212	75	14
	215	78	11
Mean	181.5	69.375	18.125
Sum	1452	555	145

	X_1^2	X_2^2	X_1y	X_2y	X_1X_2
	3600	484	8400	3080	1320
	3844	625	9610	3875	1550
	4489	576	10653	3816	1608
	4900	400	12530	3580	1400
	5041	225	13632	2880	1065
	5184	196	14400	2800	1008
	5625	196	15900	2968	1050
	6084	121	16770	2365	858
Sum	38767	2823	101895	25364	9859
Reg Sums	263.875	194.875	1162.5	-953.5	-200.375

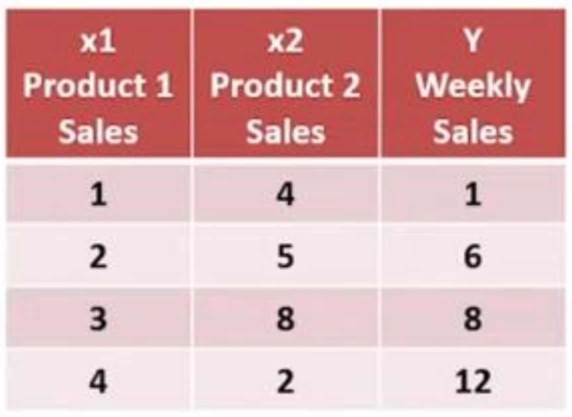
The formula to calculate w_1 is: $[(\Sigma x_2^2)(\Sigma x_1 y) - (\Sigma x_1 x_2)(\Sigma x_2 y)] \ / \ [(\Sigma x_1^2)(\Sigma x_2^2) - (\Sigma x_1 x_2)^2]$
Thus,
$\mathbf{w_1} = [(194.875)(1162.5) - (-200.375)(-953.5)] \ / \ [(263.875)(194.875) - (-200.375)^2] = \mathbf{3.148}$

The formula to calculate w_2 is: $[(\Sigma x_1^2)(\Sigma x_2 y) - (\Sigma x_1 x_2)(\Sigma x_1 y)] \ / \ [(\Sigma x_1^2)(\Sigma x_2^2) - (\Sigma x_1 x_2)^2]$
Thus,
$\mathbf{w_2} = [(263.875)(-953.5) - (-200.375)(1152.5)] \ / \ [(263.875)(194.875) - (-200.375)^2] = \mathbf{-1.656}$
The formula to calculate w_0 is: $y - w_1 X_1 - w_2 X_2$
Thus, $\mathbf{w_0} = 181.5 - 3.148(69.375) - (-1.656)(18.125) = \mathbf{-6.867}$
Step 5: Place b_0, b_1, and b_2 in the estimated linear regression equation.
The estimated linear regression equation is:
$\hat{y} = w_0 + w_1 * x_1 + w_2 * x_2$
In our example, it is $\mathbf{\hat{y} = -6.867 + 3.148x_1 - 1.656x_2}$

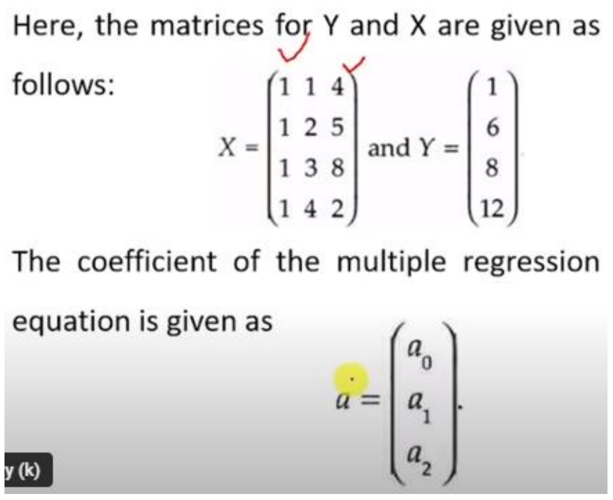
How to Interpret a Multiple Linear Regression Equation

Here is how to interpret this estimated linear regression equation:
$\hat{y} = -6.867 + 3.148x_1 - 1.656x_2$

Example :



x1 Product 1 Sales	x2 Product 2 Sales	Y Weekly Sales
1	4	1
2	5	6
3	8	8
4	2	12

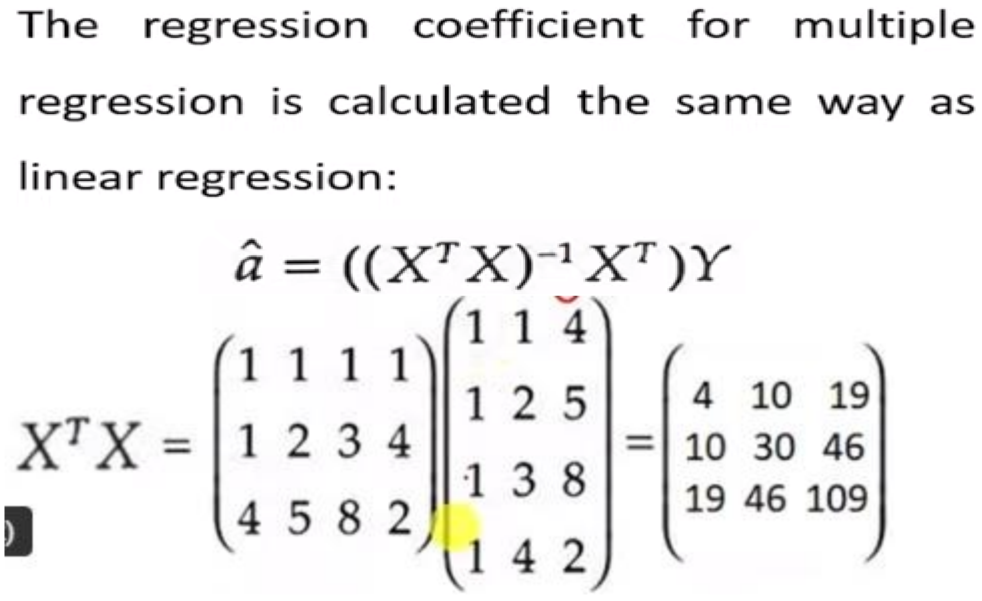


Here, the matrices for Y and X are given as follows:

$$X = \begin{pmatrix} 1 & 1 & 4 \\ 1 & 2 & 5 \\ 1 & 3 & 8 \\ 1 & 4 & 2 \end{pmatrix} \text{ and } Y = \begin{pmatrix} 1 \\ 6 \\ 8 \\ 12 \end{pmatrix}$$

The coefficient of the multiple regression equation is given as

$$a = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}.$$



The regression coefficient for multiple regression is calculated the same way as linear regression:

$$\hat{a} = ((X^T X)^{-1} X^T) Y$$

$$X^T X = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 4 & 5 & 8 & 2 \end{pmatrix} \begin{pmatrix} 1 & 1 & 4 \\ 1 & 2 & 5 \\ 1 & 3 & 8 \\ 1 & 4 & 2 \end{pmatrix} = \begin{pmatrix} 4 & 10 & 19 \\ 10 & 30 & 46 \\ 19 & 46 & 109 \end{pmatrix}$$

$$(X^TX)^{-1} = \begin{pmatrix} 4 & 10 & 19 \\ 10 & 30 & 46 \\ 19 & 46 & 109 \end{pmatrix}^{-1} = \begin{pmatrix} 3.15 & -0.59 & -0.30 \\ -0.59 & 0.20 & 0.016 \\ -0.30 & 0.016 & 0.054 \end{pmatrix}$$

$$X^TX)^{-1}X^T = \begin{pmatrix} 3.15 & -0.59 & -0.30 \\ -0.59 & 0.20 & 0.016 \\ -0.30 & 0.016 & 0.054 \end{pmatrix} \times \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 4 & 5 & 8 & 2 \end{pmatrix} = \begin{pmatrix} 0.05 & 0.47 & -1.02 & 0.19 \\ -0.32 & -0.098 & 0.155 & 0.26 \\ -0.065 & 0.005 & 0.185 & -0.125 \end{pmatrix}$$

$$\hat{a} = ((X^TX)^{-1}X^T)Y = \begin{pmatrix} 0.05 & 0.47 & -1.02 & 0.19 \\ -0.32 & -0.098 & 0.155 & 0.26 \\ -0.065 & 0.005 & 0.185 & -0.125 \end{pmatrix} \times \begin{pmatrix} 1 \\ 6 \\ 8 \\ 12 \end{pmatrix} = \begin{pmatrix} -1.69 \\ 3.48 \\ -0.05 \end{pmatrix}$$

$$a_0 = -1.69$$
$$a_1 = 3.48$$
$$a_2 = -0.05$$

- $y = a_0 + a_1x_1 + a_2x_2$
- Hence, the constructed model is:
- $y = -1.69 + 3.48x_1 - 0.05x_2$

R^2- R-squared:

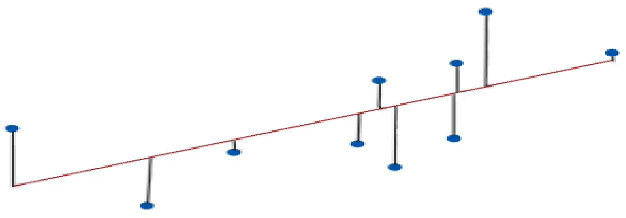
https://statisticsbyjim.com/regression/interpret-r-squared-regression/

R-squared is a goodness-of-fit measure for linear regression models. This statistic indicates the percentage of the variance in the dependent variable that the independent variables explain collectively. R-squared measures the strength of the relationship between your model and the dependent variable on a convenient 0 – 100% scale.

After fitting a linear regression model, it is need to determine how well the model fits the data. There are several key goodness-of-fit statistics for regression analysis, **R-squared** is one of them.

Linear regression identifies the equation that produces the smallest difference between all the observed values and their fitted values. Linear regression finds the smallest sum of squared residuals that is possible for the dataset. regression model fits the data well if the differences between the observations and the predicted values are small and unbiased. Unbiased means that the fitted values are not systematically too high or too low anywhere in the observation space. R-squared evaluates the scatter of the data points around the fitted regression line. It is also called the coefficient of determination, or the coefficient of multiple determination for multiple regression. For the same data set, higher R-squared values represent smaller differences between the observed data and the fitted values. R-squared is the percentage of the dependent variable variation that a linear model explains.

$$R^2 = \frac{\text{Variance explained by the model}}{\text{Total variance}}$$



Residuals are the distance between the observed value and the fitted value.

R-squared is always between 0 and 100%. Usually, the larger the R^2, the better the regression model fits the observations. Linear regression uses the sum of squares for your model to find R-squared. Consider the following formula for the given provlem.

$$R^2 = 1 - \frac{\text{sum squared regression (SSR)}}{\text{total sum of squares (SST)}} = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}.$$

Example:

y	X_1	X_2
140	60	22
155	62	25
159	67	24
179	70	20
192	71	15
200	72	14
212	75	14
215	78	11

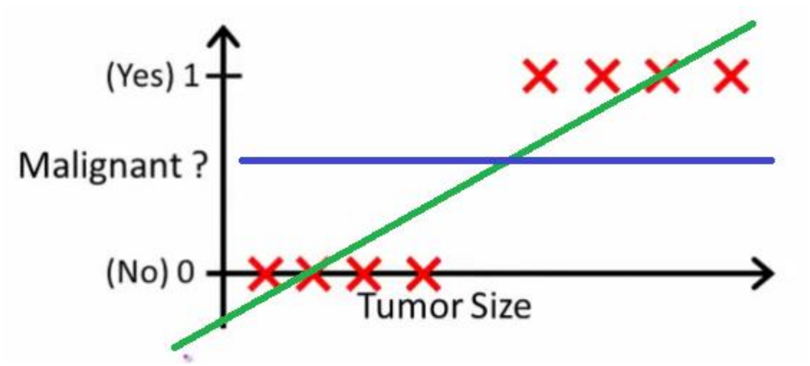
$\hat{y} = -6.867 + 3.148x_1 - 1.656x_2$

Adjusted r-squared: It can be defined as the proportion of variance explained by the model while taking into account both the number of predictor variables and the number of samples used in the regression analysis. The adjusted r-squared increases only when adding an additional variable to the model improves its predictive capability more than expected by chance alone. Adjusted R-squared is always less than or equal to R-squared.
The idea behind adjusted R-squared is to account for the addition of variables that do not significantly improve the model. When more and more predictor variables are added to the model, the R-squared will generally increase (even if those variables are only weakly associated with the response). This can give a misleading impression of improving model fit. Adjusted R-squared controls for this by penalizing the addition of uninformative predictors.
Mathematically, adjusted r-squared can be calculated as the function of R-squared in the following manner:

$$\text{Adjusted } R^2 = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1}$$ where,

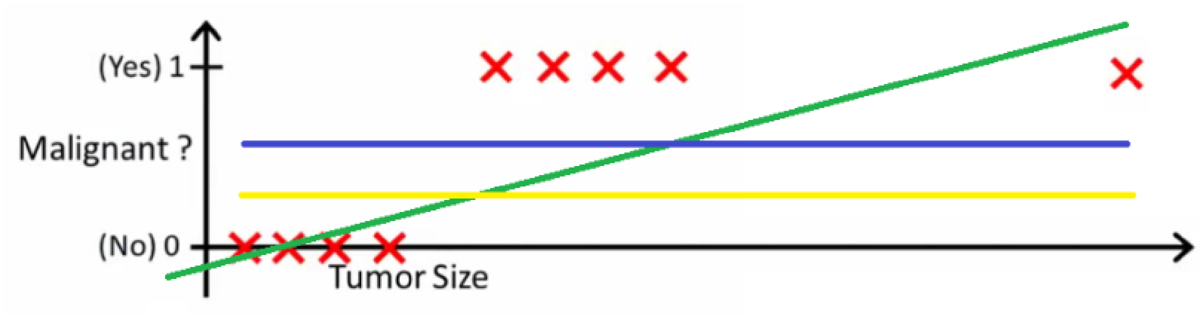
P: is the number of predictor variables.
N: is the number of records.
R^2: r-squared value of the dataset.

Logistic regression :
if we use linear regression to find the best fit line which aims at minimizing the distance between the predicted value and actual value, the line will be like this:



Here the threshold value is 0.5, which means if the value of h(x) is greater than 0.5 then we predict malignant tumour (1) and if it is less than 0.5 then we predict benign tumour (0).

If we add some outliers in our dataset, now this best fit line will shift to that point. Hence the line will be somewhat like this:

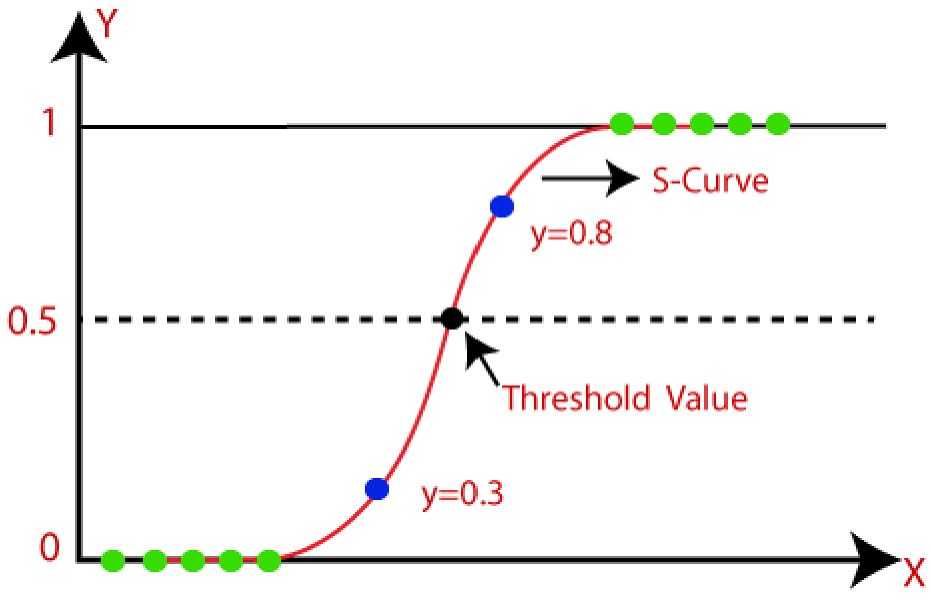


The blue line represents the old threshold and the yellow line represents the new threshold which is maybe 0.2 here. To keep our predictions right we had to lower our threshold value. Hence, we can say that linear regression is prone to outliers.
Another problem with linear regression is that the predicted values may be out of range. We know that probability can be between 0 and 1, but if we use linear regression this probability may exceed 1 or go below 0.
To overcome these problems we use Logistic Regression, which converts this straight best fit line in linear regression to an S-curve using the sigmoid function, which will always give values between 0 and 1.

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.

- Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
  The below image is showing the logistic function:



- The sigmoid function is a mathematical function used to map the predicted values to probabilities.
- It maps any real value into another value within a range of 0 and 1.
- In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

$$P = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

The three main types of logistic regression are:
1. **Binary Logistic Regression:** Used for binary classification, where the dependent variable has only two possible outcomes.
2. **Multinomial Logistic Regression:** Applied when the dependent variable has more than two categories, but they are not ordered.
3. **Ordinal Logistic Regression:** Used when the dependent variable is ordinal, meaning it has ordered categories, but the intervals between them are not necessarily equal.

Ridge Regression:
https://www.analyticsvidhya.com/blog/2016/01/ridge-lasso-regression-python-complete-tutorial/
https://vitalflux.com/ridge-regression-concepts-python-example/#google_vignette

(Ridge regression is a technique in machine learning that addresses the issue of overfitting in linear models. In linear regression, we aim to model the relationship between a response variable and one or more predictor variables. However, when there are multiple variables that are highly correlated, the model can become too complex and prone to overfitting. This is where ridge regression comes into play).

The ridge regression is a type of linear regression model (~~that aids in analyzing multicollinearity in multiple regression data~~). It aims to reduce the sum of squared errors between the actual and predicted values and actual values by adding a penalty term that diminishes the coefficients and brings them closer to zero.

Ridge regression in machine learning decreases the **standard error** by adding a penalty term to the **regression** approximations. It aids in getting more accurate estimates. This regression performs L2 regularization by penalizing the weights of the feature's coefficients and decreasing the value between the actual and predicted observations. Furthermore, it prevents overfitting and reduces the model's complexity. It is especially beneficial when the data set contains a more significant number of predictors than the number of observations.

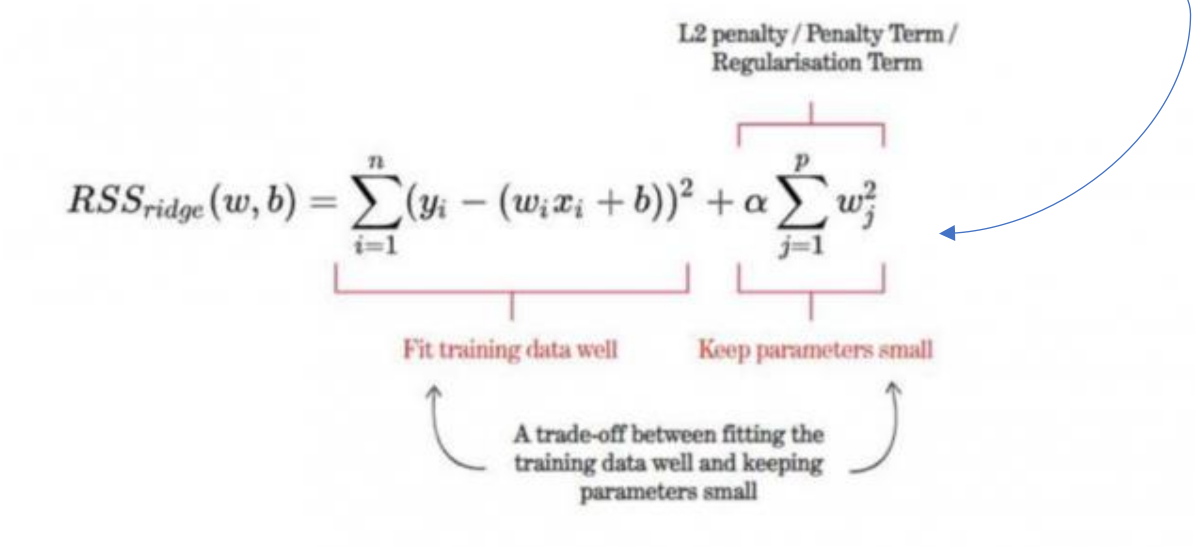
The ridge regression formula is:

min RSS + α * ||β||2

Where RSS = **residual sum of squares** which is the sum of squared differences between the predicted and actual values

β = weights of the coefficients of the **independent variables**

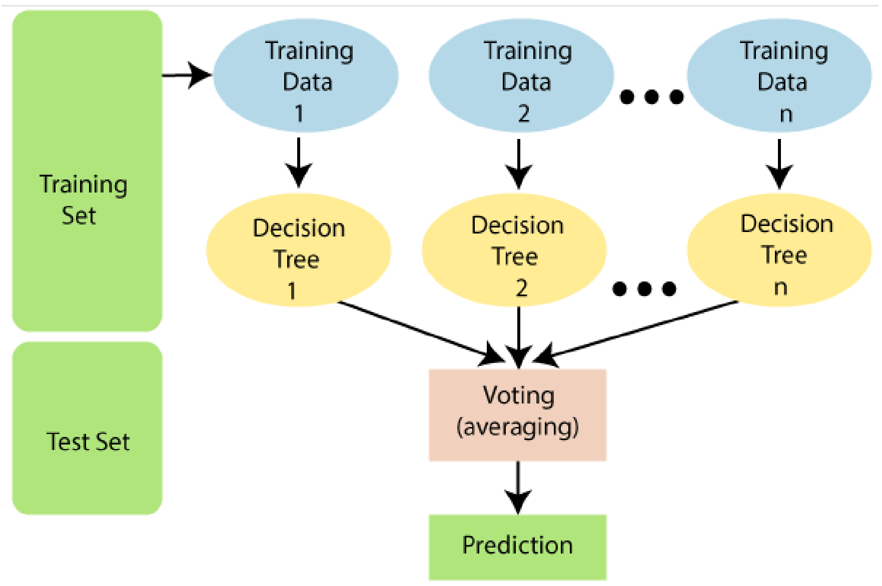
α = a regularization parameter that controls the strength of the penalty term


L2 penalty / Penalty Term / Regularisation Term
$RSS_{ridge}(w,b) = \sum_{i=1}^{n}(y_i - (w_i x_i + b))^2 + \alpha \sum_{j=1}^{p} w_j^2$
Fit training data well
Keep parameters small
A trade-off between fitting the training data well and keeping parameters small

Random Forest Algorithm:

Random Forest algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model.*

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. **The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.**



Random Forest works in two-phases first is to create the random forest by combining N decision trees, and second is to make predictions for each tree created in the first phase.

Steps Involved in Random Forest Algorithm

- **Step 1:** In the Random forest model, a subset of data points and a subset of features is selected for constructing each decision tree.
- **Step 2:** Individual decision trees are constructed for each sample.
- **Step 3:** Each decision tree will generate an output.
- **Step 4:** Final output is considered based on ***Majority Voting or Averaging*** for Classification and regression, respectively.

K-Nearest Neighbor) Algorithm:

The K-Nearest Neighbor (KNN) algorithm is used for classification and regression tasks. It works on the idea that similar data points tend to have similar labels or values.
During the training phase, the KNN algorithm stores the entire training dataset as a reference. When making predictions, it calculates the distance between the input data point and all the training examples, using a chosen distance metric such as Euclidean distance.
Next, the algorithm identifies the K nearest neighbors to the input data point based on their distances. In the case of classification, the algorithm assigns the most common class label among the K neighbors as the predicted label for the input data point. For regression, it calculates the average or weighted average of the target values of the K neighbors to predict the value for the input data point. Step by step procedure is given bellow.

1. Decide on your similarity or distance metric.
2. Split the original labelled dataset into training and test data.
3. Pick an evaluation metric. (Misclassification rate is a good one)
4. Run k-NN a few times, changing k and checking the evaluation measure.
5. Optimize k by picking the one with the best evaluation measure.
6. Once you've chosen k, use the same training set and now create a new test set that you have no labels that want to predict.

Similarity measures:

The dissimilarity (or similarity) between the objects described by interval-scaled variables is typically computed based on the distance between each pair of objects. The most popular distance measure is Euclidean distance.
which is defined as

$$d(i,j)=\sqrt{(x_{i1}-x_{j1})^2+(x_{i2}-x_{j2})^2+\cdots+(x_{in}-x_{jn})^2},$$

where $i=(x_{i1},x_{i2},\ldots,x_{in})$ and $j=(x_{j1},x_{j2},\ldots,x_{jn})$ are two n-dimensional data objects.
Manhattan (or city block) distance, defined as

$$d(i,j)=|x_{i1}-x_{j1}|+|x_{i2}-x_{j2}|+\cdots+|x_{in}-x_{jn}|.$$

Minkowski distance is a generalization of both Euclidean distance and Manhattan distance. It is defined as

$$d(i,j)=(|x_{i1}-x_{j1}|^p+|x_{i2}-x_{j2}|^p+\cdots+|x_{in}-x_{jn}|^p)^{1/p},$$

Cosine similarity is a metric used to measure the similarity of two vectors. Specifically, it measures the similarity in the direction or orientation of the vectors ignoring differences in their magnitude or scale. The similarity of two vectors is measured by the cosine of the angle between them. if we have two vectors, A and B, the similarity between them is calculated as:

$$similarity(A,B)=cos(\theta)=\frac{A\cdot B}{\|A\|\|B\|}$$

where

- θ is the angle between the vectors,
- $A\cdot B$ is dot product between A and B and calculated as $A\cdot B=A^TB=\sum_{i=1}^{n}A_iB_i=A_1B_1+A_2B_2+\ldots+A_nB_n$,
- $\|A\|$ represents the L2 norm or magnitude of the vector which is calculated as $\|A\|=\sqrt{A_1^2+A_1^2\ldots A_1^n}$.

Example:

- $D1=[1,1,1,1,1,0,0]$
- $D2=[0,0,1,1,0,1,1]$

$$D1\cdot D2=1\times0+1\times0+1\times1+1\times1+1\times0+0\times1+0\times1=2$$

$$\|D1\|=\sqrt{1^2+1^2+1^2+1^2+1^2+0^2+0^2}=\sqrt{5}$$

$$\|D2\|=\sqrt{0^2+0^2+1^2+1^2+0^2+1^2+1^2}=\sqrt{4}$$

$$similarity(D1,D2)=\frac{D1\cdot D2}{\|D1\|\|D2\|}=\frac{2}{\sqrt{5}\sqrt{4}}=\frac{2}{\sqrt{20}}=0.44721$$

Jaccard Distance or Similarity:

Jaccard Similarity is a common proximity measurement used to compute the similarity between two objects, such as two text documents

The Jaccard Similarity can be used to compute the similarity between two asymmetric binary variables (the two states 0,1 are not equally important).

- a = the number of attributes that equal 1 for both objects i an j
- b = the number of attributes that equal 0 for object i but equal 1 for object j
- c = the number of attributes that equal 1 for object i but equal 0 for object j
- d = the number of attributes that equal 0 for both objects i and j.

Then, Jaccard Similarity for the attributes I and j is calculated by the following equation:

$$J(i,j) = sim(i,j) = \frac{a}{a+b+c}$$

Example:

	item1	item2	item2	item4	item5	item6	item7	item8	item9
C1	0	1	0	0	0	1	0	0	1
C2	0	0	1	0	0	0	0	0	1
C3	1	1	0	0	0	1	0	0	0

$$J(C1,C2) = \frac{a}{a+b+c} = \frac{1}{1+1+2} = 0.25 J(C1,C3) = \frac{a}{a+b+c} = \frac{2}{2+1+1} = 0.5$$

The dissimilarity by the Jaccard Coefficient is computed as follows:

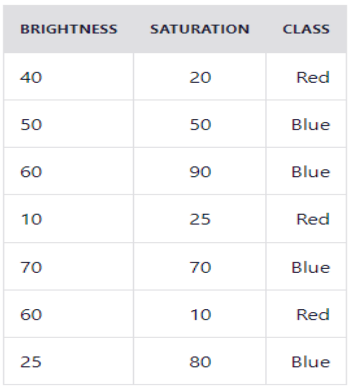
$$d(i,j) = \frac{b+c}{a+b+c} \implies 1 - sim(i,j)$$

Jaccard Similarity for Two Sets:

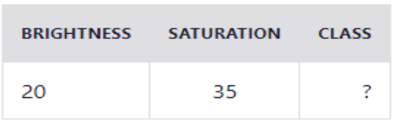
The Jaccard similarity measures the similarity between two sets of data to see which members are shared and distinct. Let A and B are two sets then,

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Example: Consider the following database and new data and assume the value of **K** is 5.

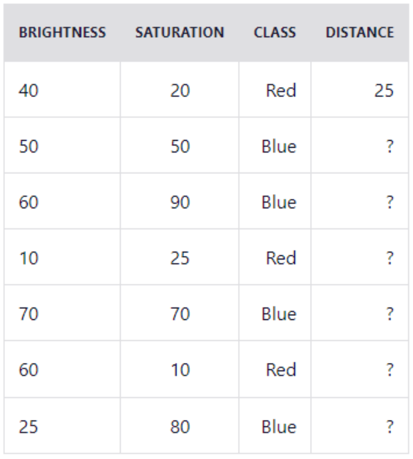


BRIGHTNESS	SATURATION	CLASS
40	20	Red
50	50	Blue
60	90	Blue
10	25	Red
70	70	Blue
60	10	Red
25	80	Blue

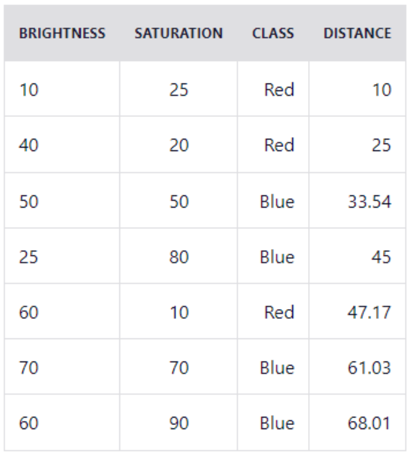


BRIGHTNESS	SATURATION	CLASS
20	35	?

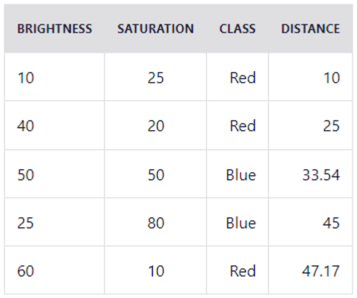
To know its class, we have to calculate the distance from the new entry to other entries in the data set using the Euclidean distance formula. Here's the formula: $\sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$



BRIGHTNESS	SATURATION	CLASS	DISTANCE
40	20	Red	25
50	50	Blue	?
60	90	Blue	?
10	25	Red	?
70	70	Blue	?
60	10	Red	?
25	80	Blue	?



BRIGHTNESS	SATURATION	CLASS	DISTANCE
10	25	Red	10
40	20	Red	25
50	50	Blue	33.54
25	80	Blue	45
60	10	Red	47.17
70	70	Blue	61.03
60	90	Blue	68.01



BRIGHTNESS	SATURATION	CLASS	DISTANCE
10	25	Red	10
40	20	Red	25
50	50	Blue	33.54
25	80	Blue	45
60	10	Red	47.17

Mahalanobis Distance:

Also, can be used between two real-valued vectors and has the advantage over Euclidean distance that it considers correlation and scale-invariant.

$$d(\vec{x},\vec{y})=\sqrt{(\vec{x}-\vec{y})^T S^{-1}(\vec{x}-\vec{y})},$$

where S is the covariance matrix.

Hamming Distance:

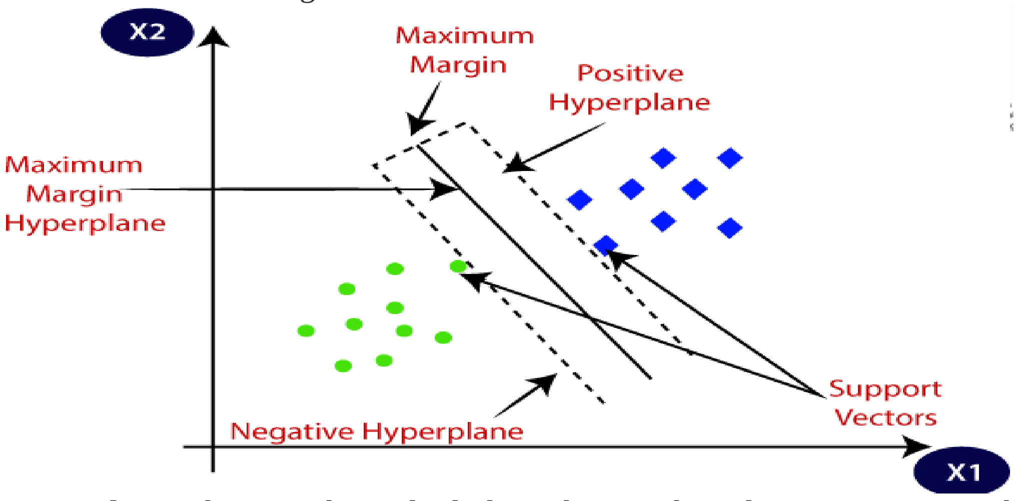
It Can be used to find the distance between two strings or pairs of words or DNA sequences of the same length.

The distance between olive and ocean is 4, because aside from the "o" the other 4 letters are different. The distance between shoe and hose is 3 be- cause aside from the "e" the other 3 letters are different. Here we just go through each position and check whether the letters the same in that position, and if not, increment the count by 1.

Support Vector Machine:

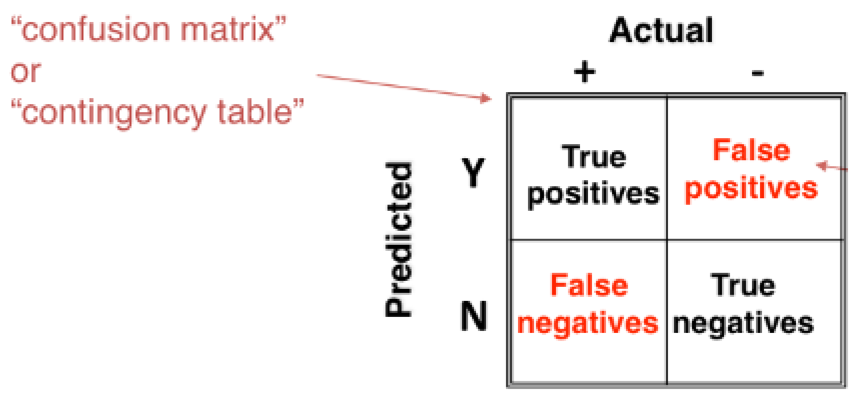
Support Vector Machine (SVM) is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space

into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.
SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram:



Hyperplane: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.
We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.
Support Vectors: The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.
Note: the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a **hyperplane**. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as **margin**. And the goal of SVM is to maximize this margin. The **hyperplane** with maximum margin is called the **optimal hyperplane**.

Confusion Matrix:
It gives the matrix as output and describes the complete performance of the model. Lets assume we have a binary classification problem. We have some samples belonging to two classes : YES or NO. Also, we have our own classifier which predicts a class for a given input sample.



There are 4 important terms :

- **True Positives**: The cases in which we predicted YES and the actual output was also YES.
- **True Negatives**: The cases in which we predicted NO and the actual output was NO.
- **False Positives**: The cases in which we predicted YES and the actual output was NO.
- **False Negatives**: The cases in which we predicted NO and the actual output was YES.

Accuracy:
It is the ratio of number of correct predictions to the total number of input samples.

$$\text{Accuracy} = \frac{\text{True positives} + \text{True negatives}}{\text{True positives} + \text{False positives} + \text{True negatives} + \text{False negatives}}$$

It works well only if there are equal number of samples belonging to each class (Balanced database).

(For example, consider that there are 98% samples of class A and 2% samples of class B in our training set. Then our model can easily get **98% training accuracy** by simply predicting every training sample belonging to class A.
When the same model is tested on a test set with 60% samples of class A and 40% samples of class B, then the **test accuracy would drop down to 60%.** Classification Accuracy is great, but gives us the false sense of achieving high accuracy).

Confusion Matrix forms the basis for the other types of metrics.

- **true positive rate** = TP/(TP+FN) = 1 − false negative rate
- **false positive rate** = FP/(FP+TN) = 1 − true negative rate
- **sensitivity** = true positive rate
- **specificity** = true negative rate
- **positive predictive value** = TP/(TP+FP)
- **recall** = TP / (TP+FN) = true positive rate
- **precision** = TP / (TP+FP)

- **F-score** is the harmonic mean of precision and recall:
  $F = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$
- **G-score** is the geometric mean of precision and recall:
  $G = \sqrt{\mathrm{precision} \cdot \mathrm{recall}}$