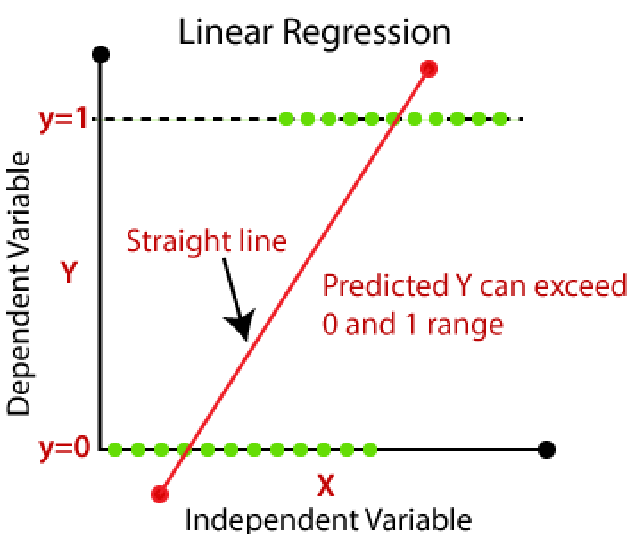
Logistic Regression is one of the most simple and commonly used Machine Learning algorithms for two-class classification. I.e **binary class**.
It is a special case of linear regression where the target variable is categorical in nature. It uses a log of odds as the dependent variable. Logistic Regression predicts the probability of occurrence of a binary event utilizing a logit function.



A linear line needs to be clipped at 0 and 1 (a). So that it looks like (b), a sigmoid curve.
We need to create a threshold value:
Ex: if, x>0.5, then 1
x<0.5, then 0
Hence we need a function that can give output 0 or 1.
We can use the odd ratio, the probability of success divided by the probability of failure.
S=P/1-P --- **Equation of odd ratio**
Where, P=1/1+e^(-u) --- **A sigmoid function**
u= B0 + B1X1 ------ **A Linear Regression Equation (where a cost function can be added)**

But, In the Equation of odd ratio:
If, P=0 then S=0 and P=1 then S= Infinity.
So, the range is between 0 to infinity. But we want a range from - infinity to + infinity.
Hence Applying a log to odd ratio. Making a **logit function**: Logit [P/1-P]

$$S = Odds\ ratio = \frac{p}{1-p}$$

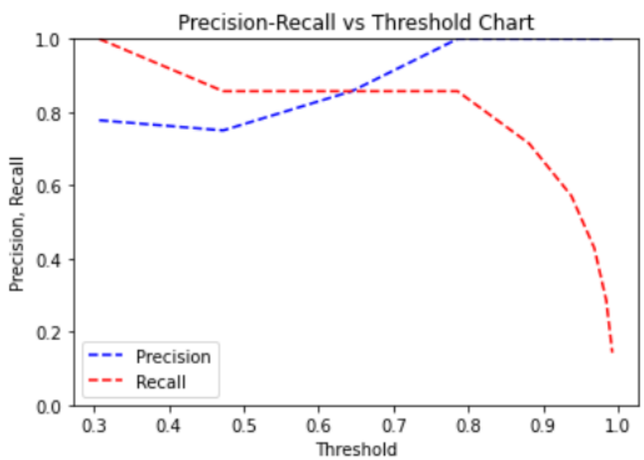
$$S = \frac{\frac{e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k}}{1 + e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k}}}{1 - \frac{e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k}}{1 + e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k}}}$$

$$\therefore, S = e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k}$$

$$\ln(S) = \ln\left(e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$$ **- Logit function**

If the curve goes to positive infinity, y predicted will become 1, and if the curve goes to negative infinity, y predicted will become 0. If the output of the sigmoid function is more than 0.5, we can classify the outcome as 1 or YES, and if it is less than 0.5, we can classify it as 0 or NO.

Precision-Recall Curve can be used to get the best threshold in the case of binary prediction.
Ex : `precision,recall,thresholds = precision_recall_curve(y, probs_y[:, 1])`



Advantages of a log: 1. It maximizes the likelihood by normalizing the coefficients.
2. Easier operation such as - + x %
3. Creates a Curve/Hyperplane enable to choose smaller or larger coefficients.
Properties of Logistic Regression:
The dependent variable in logistic regression follows Bernoulli Distribution.
Estimation is done through maximum likelihood.
Note: In linear Regression, we minimize SSE but in Logistic Regression, we maximise log-likelihood.
• Log Likelihood is a convex function and hence finding optimal parameters is easier.

$$\text{Log Likelihood} = \sum[\, y_i \ln \hat{y}_i \; + \; (1 - y_i)\ln(1 - \hat{y}_i)]$$

Forward Steps:
To maximize likelihood we use the cost function. where we use the same principle of simple linear regression and find the best possible value with the lowest sum of the square error to get close to accuracy.
To optimize this cost function, we take the likelihood function of both success and non-success and do the partial derivation (apply gradient descent). This helps us to better predict the most likely value of p-success in the equation.
To optimize it more we can further go for the stochastic gradient where we take joint likelihood to predict the best possible value of p.
Cost function intuition
If the actual class is 1 and the model predicts 0, we should highly penalize it and vice-versa.

Reference link for cost function:
https://towardgistic-regression-in-python-301d27367csdatascience.com/building-a-lo24

Example:

Suppose we want a probability that a 50-year old club member will return the form.

$$\ln(S) = -20.40782 + 0.42592 * 50 = 0.89$$
$$S = e^{0.89} = 2.435$$

S=p/1-p

HENCE

$$\hat{p} = \frac{S}{S+1} = \frac{2.435}{2.435+1} = 0.709$$

Using a probability of 0.50 as a cut-off between predicting a 0 or a 1, this member would be classified as a 1.