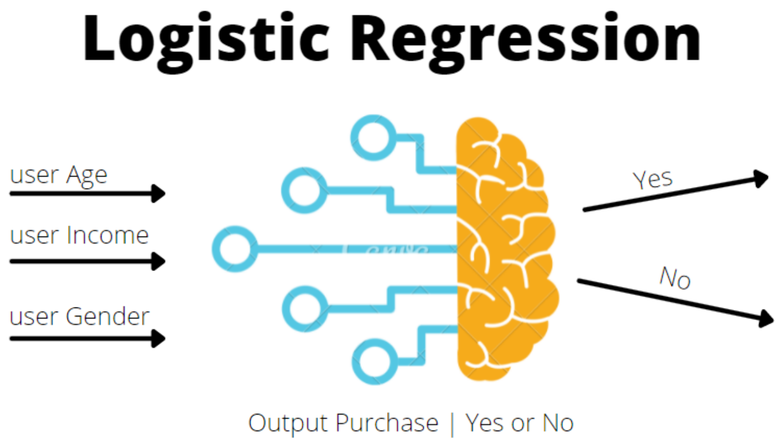
<u>Logistic Regression</u>:

Logistic Regression is used when we have output (dependent variable ŷ) as categorical.

It is a statistical analysis to predict binary outputs, such as “true/false” or “yes/no”, based on observations and the outputs can be in between 0 and 1. Therefore, we can say that logistic regression acts as a binary classifier.



Connect with me:

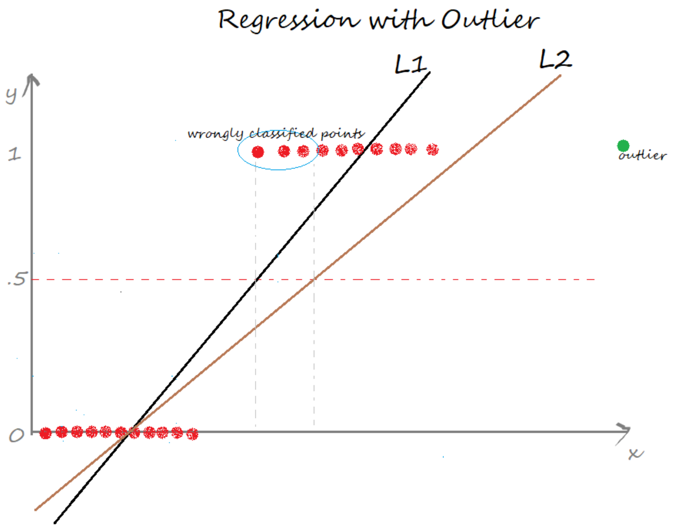
Linked in - https://www.linkedin.com/in/sai-subhasish-rout-655707151/

Github - https://github.com/saisubhasish/Concepts

Why we can’t use the linear regression to the classification problems?

Because of outliers in the dataset, the line will get switched and the output of our model will be wrong with respect to threshold.

Another reason is that there is no limit to the prediction of linear regression model and in logistic regression there is no sense of the values greater than 1 or less than 0. That’s why we use logistic regression where the output ranges between 0 and 1.



We need a function we transform the values in such a way that the output will range between 0 and 1. And we try to squash the best fit line in between 0 and 1 and make it straight.

Sigmoid Function:

Sigmoid function maps a real value with any range into another real value between 0 and 1. So here we can use probability by setting the threshold.

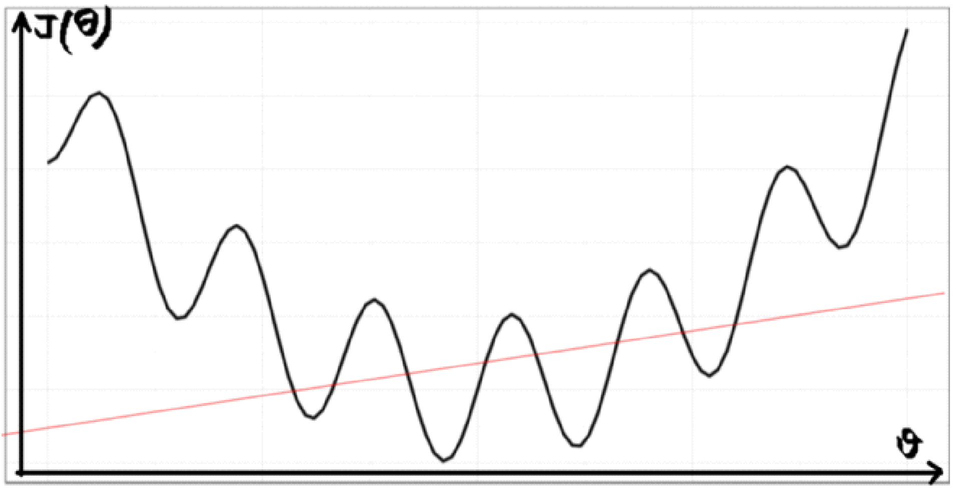
Threshold can be calculated with the help of ROC and AOC curve.

Y = Q(Z)

Q(Z) = 1 / (1 + e^{-x})

$$h\theta(X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

To measure the performance of the logistic regression we don't use MSE, because if we will apply ŷ = 1 / (1 + e^{-x}) to MSE we will get non-convex curve.



To fix this issue we use log loss function.

Cost function → Log loss

Using cost function we get an optimize value of a model where error will be minimum.

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$\mathrm{Cost}(h_\theta(x), y) = -\log(h_\theta(x)) \qquad \text{if } y = 1$$

$$\mathrm{Cost}(h_\theta(x), y) = -\log(1 - h_\theta(x)) \qquad \text{if } y = 0$$

We also can write the function as

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)})$$

$$J(\theta) = \frac{1}{m}[\sum_{i=1}^{m} -y^{(i)} log(h_\theta(x^{(i)})) + (1 - y^{(i)}) log(1 - h_\theta(x^{(i)}))]$$

$$m = number\ of\ samples$$

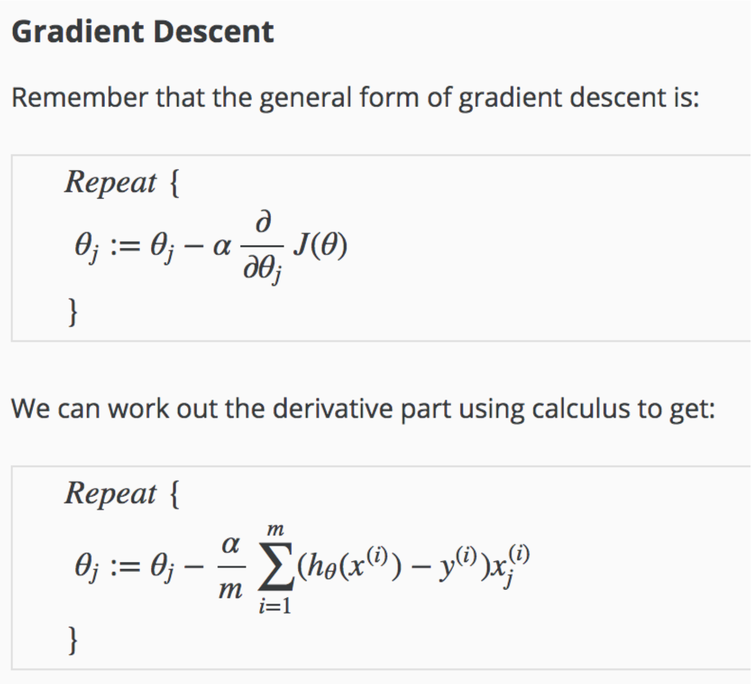
By using this formula we will never get local minima.

Log loss

Log loss is a classification metric calculated based on probabilities. It is a good metric to compare models. A lower loss value means better accuracy or prediction.

Gradient Descent :

We use Gradient Descent to minimize cost function



Gradient Descent

Remember that the general form of gradient descent is:

$$Repeat\ \{$$
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$
$$\}$$

We can work out the derivative part using calculus to get:

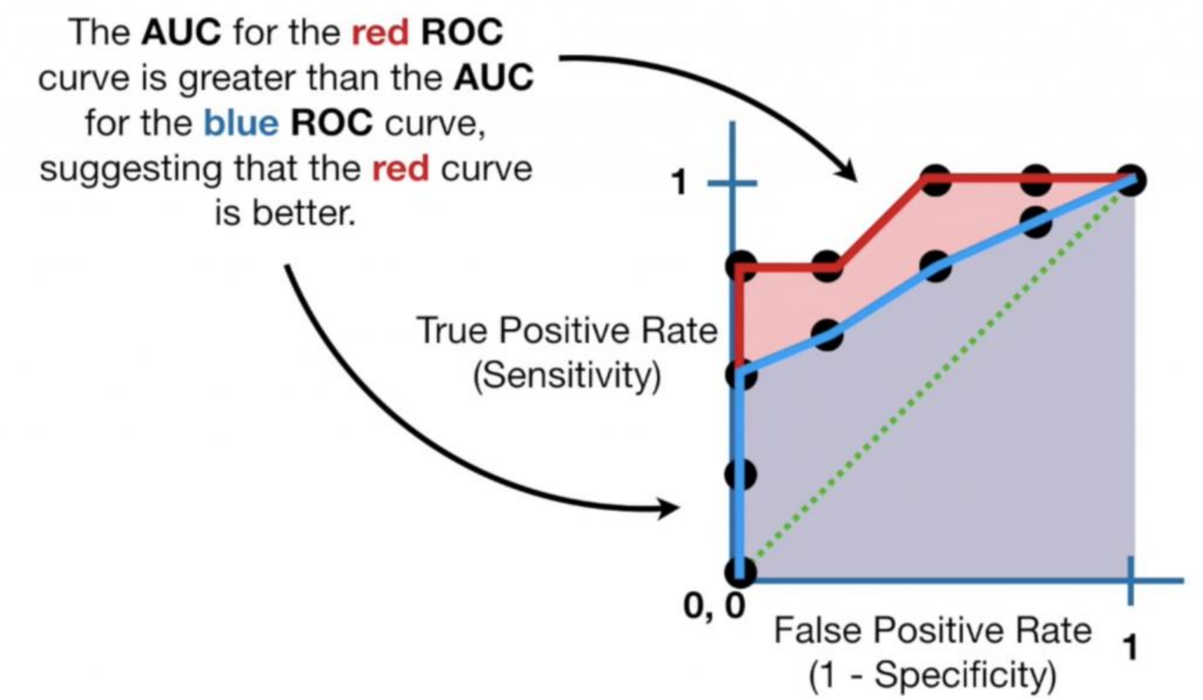
$$Repeat\ \{$$
$$\theta_j := \theta_j - \frac{\alpha}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$
$$\}$$

Every time we update the value of θj.

ROC-AUC curve :

An **ROC curve** (**receiver operating characteristic curve**) is a graph which shows the performance of a model at different classification thresholds.

Using ROC we will get the thresholds and threshold is being selected by domain expertise.



AUC(**Area Under the Curve**) is the area ROC(threshold) is covering. The higher the area of AUC the better the model is.

ROC curve plots two parameters:

- False Positive Rate
- True Positive Rate

True Positive Rate (**TPR**):

TPR=TP/(TP+FN)

False Positive Rate (**FPR**) is defined as follows:

FPR=FP/(FP+TN)

Variance inflation factor (VIF):

Variance Inflation Factor measures to determine which variables are highly correlated after interaction with other independent variables.

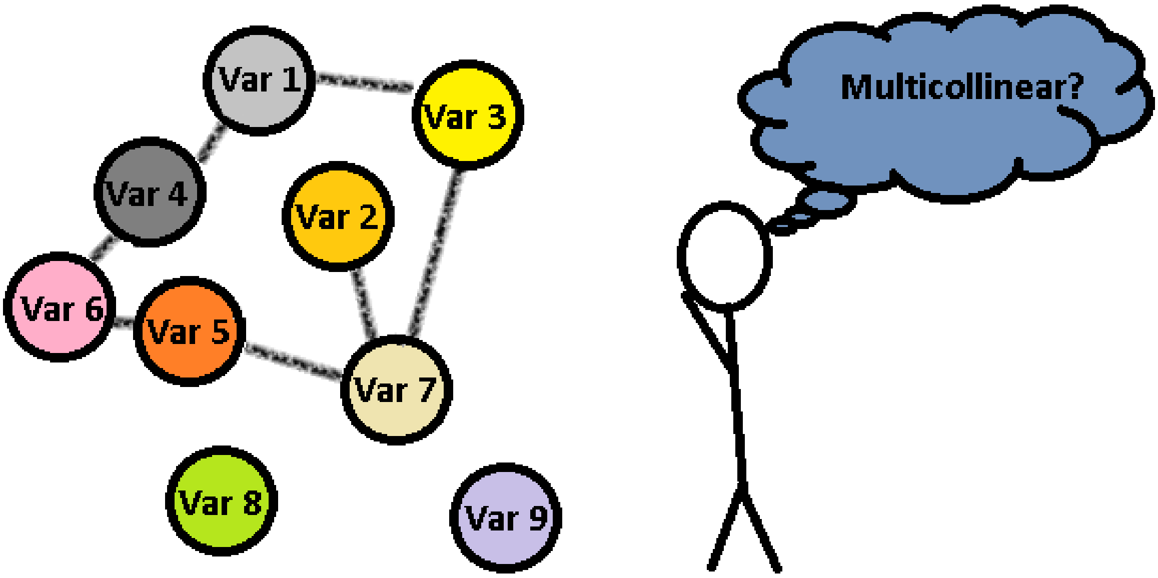
In simple language the behaviour of a feature with respect other features (similar to model prediction).

Features with high VIF score will get removed.

The variables with VIF score 4 to 5 is considered as moderate to high and with score 10 is considered as very high.

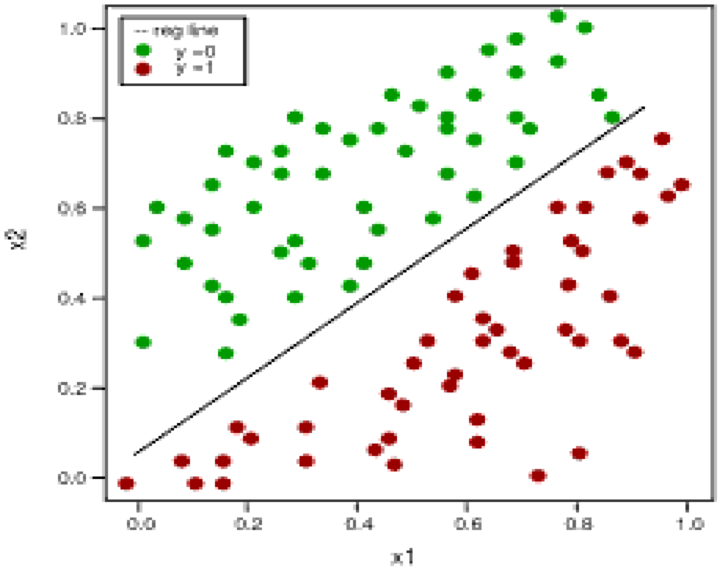
$$VIF_i = \frac{1}{1 - R_i^2}$$

Here R-square represents multicollinearity which is produced by model. High value of R-square indicates higher multicollinearity.



Performance Metrics : (classification problem)

- Confusion Matrix
- Accuracy
- Precision
- Recall
- F-Beta Score



Confusion Matrix :

Confusion Matrix helps us to visualize the outcome of Logistic Regression in a tabular form. There will be all the predicted and actual value of model and dataset.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Accuracy :

This metrics tell us that how many are correctly predicted from total predictions.

We calculate accuracy by:

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

Because of imbalance dataset we may get high accuracy. So measuring accuracy is not sufficient.

Precision :

Out of all actual values how many are correctly predicted.

For an Example – If we are predicting for positive outputs(200), then how from them correctly predicted (150), and how many of them are incorrect(50).

Precision = TruePositives / (TruePositives + FalsePositives)

Precision = 150 / (150 + 50) = 75%

If you have imbalance dataset where FP is important then you have to focus on ‘Precision’.

$$\textbf{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

$$\textbf{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

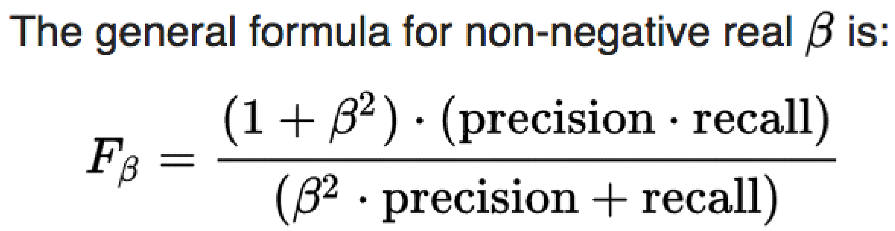
Recall :

Out of all the predicted values, how many are correctly predicted.

Means how many individuals are correctly predicted out of all positive predictions.

If we have imbalance dataset where FN is important then we have to focus on Recall.

F-Beta Score :



The general formula for non-negative real β is:

$$F_\beta = \frac{(1+\beta^2)\cdot(\mathrm{precision}\cdot\mathrm{recall})}{(\beta^2\cdot\mathrm{precision}+\mathrm{recall})}$$

F-Beta score considers both precision and recall to compute the score. It is the weighted average of precision and recall. If both precision and recall score is high then F-Beta score will be high. If the F-Beta score of you model is high then you can consider your model’s performance is well.

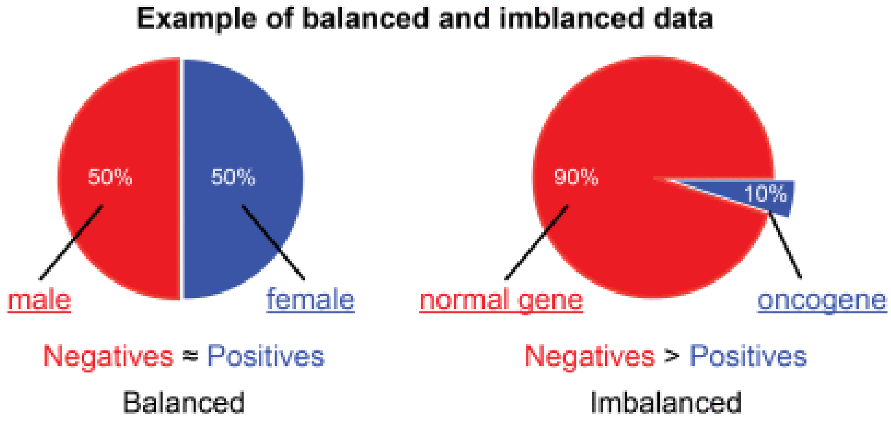
Imbalanced Dataset :

In classification problem we get output in labels. There we face imbalanced dataset problem. Where the output of one class is comparatively low with compared to other class in binary classification problem.

So imbalance dataset refers to un-even distribution of classes within dataset.

Example :

1. Credit-card fraud detection
2. Natural Disaster
3. Dieses Diagnosis

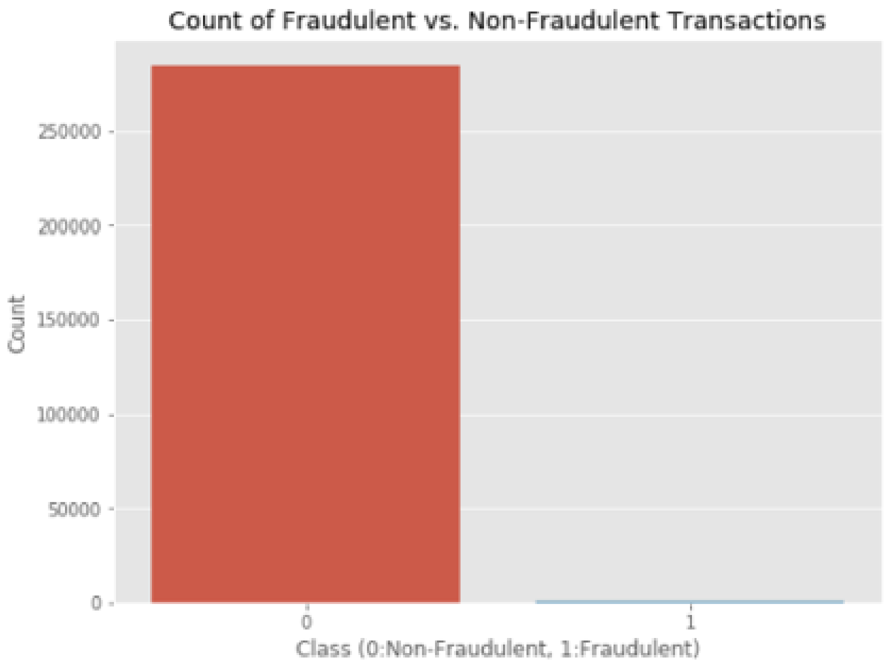


Random Sampling (Under sampling & Over Sampling):

In imbalance dataset for classification problem we face un-even(skewed) distribution for the output classes.

This way it can affect our machine learning model algorithm, as it may completely ignore the minority classes. But in some of the cases these minority classes are most important for us to do prediction.

eg: credit card fraud detection



Here if the two classes were equally distributed then that is also a problem.

So, to handle this kind of situation we have an approach called Random Sampling. There are two ways to perform random sampling

1. Over sampling (It makes duplicates of the minority class)
2. Under sampling (It deletes the samples from majority class)

Random sampling creates a new transformed version of the data where there will be a new class distribution to reduce the influence of the data.

Thank you

Hope you liked the Post

Connect with me to have the intuition behind the concepts