Machine Learning to build Intelligent Systems

Manas Dasgupta

Understanding Logistic Regression

Structure of this Module

Understanding Logistic Regression through a Classification Problem (Project)

TOPICS
Introduction to Logistic Regression
Estimating Probabilities
Logistic Regression Cost Functions
Softmax Regression
Performance Metrics
ROC Curve and AUC
Optimising Logistic Regression Model

Understanding the Logit Model

- In statistics, the **logistic model** (or **logit model**) is used to ***model the probability of a certain class or event*** existing such as pass/fail, win/lose, alive/dead or healthy/sick.

- Logistic regression is a statistical model that in its basic form uses a ***logistic function to model a binary dependent variable***. In regression analysis, **logistic regression** (or **logit regression**) is **estimating the parameters of a logistic model** (a form of binary regression).

- Mathematically, a **binary logistic model** has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labelled "0" and "1". In the logistic model, the **log-odds (the logarithm of the odds)** for the value labelled "1" is a linear combination of one or more independent variables ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value).

- The corresponding probability of the value labelled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), hence the labelling; the **function that converts log-odds to probability is the logistic function**.

- The defining characteristic of the logistic model is that increasing one of the independent variables multiplicatively scales the odds of the given outcome at a *constant* rate.

- Outputs with more than two values are modelled by **multinomial logistic regression** and, if the multiple categories are ordered, by **ordinal logistic regression** (for example the proportional odds ordinal logistic model).

Understanding the Logit Model

Let us try to understand logistic regression by considering a logistic model with given parameters, then seeing how the coefficients can be estimated from data. Consider a model with two predictors, x1 and x2, and one binary (Bernoulli) response variable Y, which we denote p = P(Y = 1).

We assume a linear relationship between the predictor variables and the log-odds (also called logit) of the event that Y = 1. This linear relationship can be written in the following mathematical form (where l, is the log-odds, b is the base of the logarithm, and β are parameters of the model).

$$\ell = \log_b \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

Log Odds

$$\frac{p}{1-p} = b^{\beta_0+\beta_1 x_1+\beta_2 x_2}$$

Odds

$$p = \frac{b^{\beta_0+\beta_1 x_1+\beta_2 x_2}}{b^{\beta_0+\beta_1 x_1+\beta_2 x_2}+1} = \frac{1}{1+b^{-(\beta_0+\beta_1 x_1+\beta_2 x_2)}} = S_b(\beta_0 + \beta_1 x_1 + \beta_2 x_2)$$

Here, S_b is the Sigmoid Function with base b.
The formula on the left shows that once β_i is fixed, e can easily compute either the log-odds that Y = 1 or a given observation.

The main use-case of a logistic model is to be given an observation (x1, x2) and estimate the probability p, that Y = 1. In most applications, the base b, of the logarithm is usually taken to be *e* (Euler number, equivalent to 2.71828).

Understanding the Logit Model

The **logistic function** is a **sigmoid function**, which takes any real input *t*, and **outputs a value between zero and one**. For the **logit**, this is interpreted as taking input **log-odds and having output probability**.

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

Here, *t* is a linear function of a single explanatory variable *x, i.e., t* is a *linear combination* of multiple explanatory variables is treated similarly.

$$t = \beta_0 + \beta_1 x$$

The general logistic function can be written as:

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

Has output Probability

Understanding the Logit Model

We consider an example with b (log-base) = 10, and coefficients β_0 = -3, β_1 = 1, β_2 = 2.

The model will read as:

$$\log_{10} \frac{p}{1-p} = \ell = -3 + x_1 + 2x_2$$

[Where *p* is the probability of the event that Y = 1]

This can be interpreted as follows:

- β_0 is the y-intercept. It is the log-odds of the event that Y = 1, when the predictors x1 = x2 = 0.
- When β_1 = 1, increasing x1 by 1 increases the log-odds for Y = 1 by 1, i.e., the odds increase by a factor of 10^1. Note that the **probability** of Y = 1 has also increased, but it has not increased by as much as the odds have increased.
- When β_2 = 2, increasing x2 by 1 increases the log-odds for Y = 1 by 2, i.e., the odds increase by a factor of 10^2. Note how the effect of x2 on the log-odds is twice as great as the effect of x1, but the effect on the odds is 10 times greater. But the effect on the **probability** of Y = 1 is not as much as 10 times greater, it's only the effect on the odds that is 10 times greater.

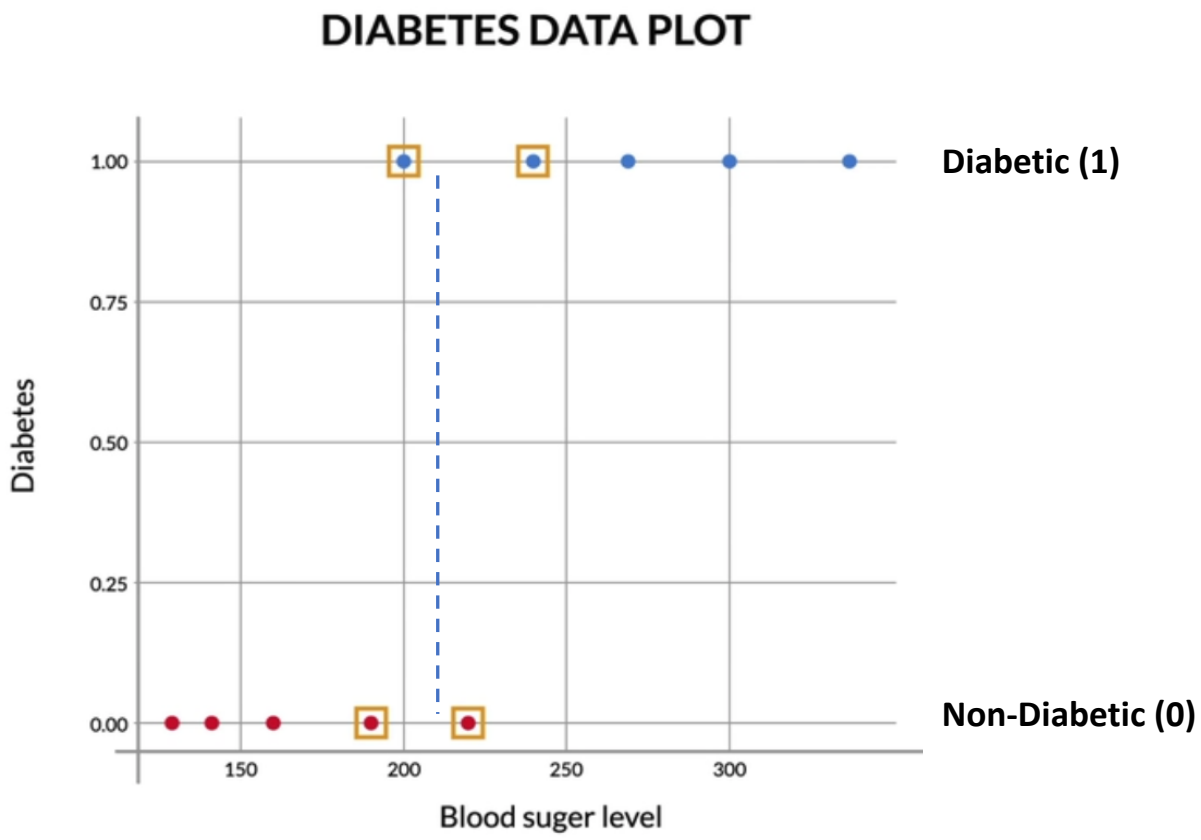
Introduction to Logistic Regression

Let us look at an example of determining whether or not a person has diabetes based on Blood Sugar level reading.

DIABETES DATA

Blood Sugar Level	Diabetes
190	No
240	Yes
300	Yes
160	No
200	Yes
269	Yes
129	No
141	No
220	No
337	Yes

Problem statement:
Given a Blood Sugar value, say 210, what is the probability of Diabetes being 1?



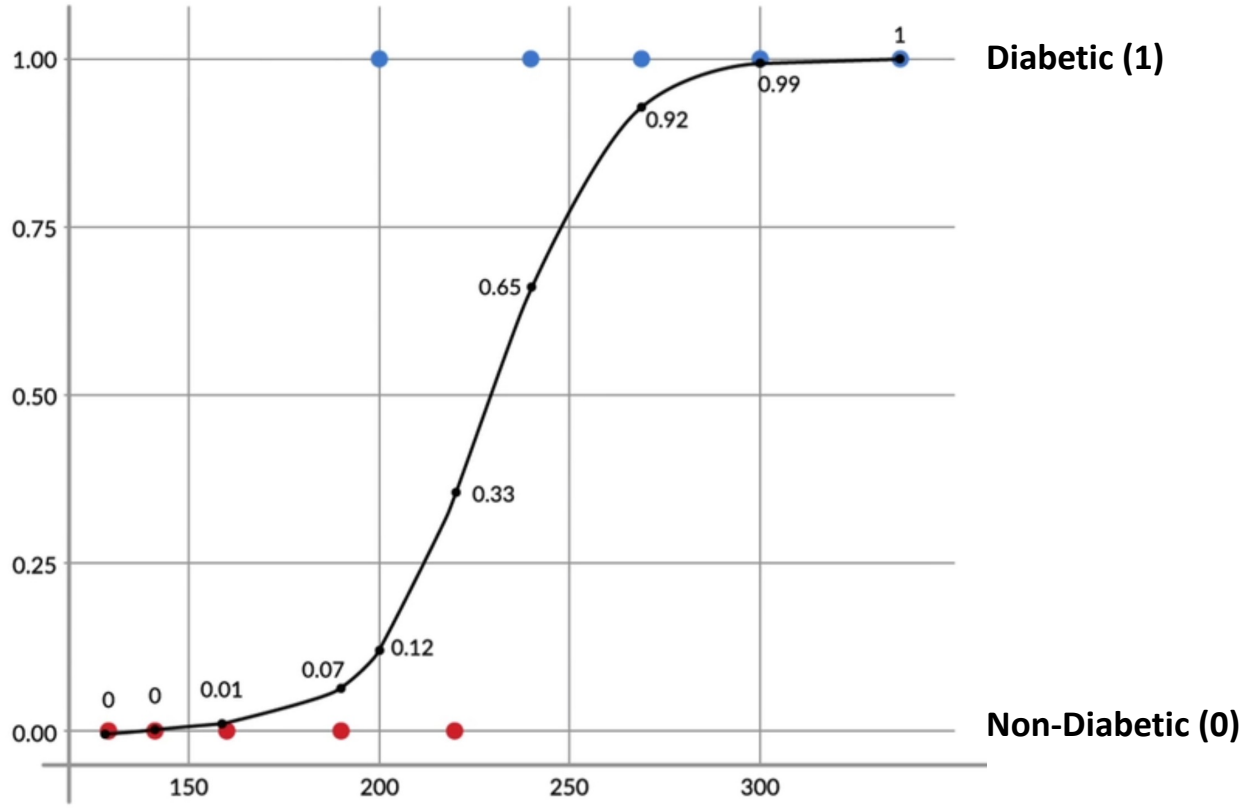
Introduction to Logistic Regression

Plotting the Probabilities

Sigmoid Curve

Sigmoid Function

$$y\ (\text{Probability of Diabetes}) = \frac{1}{1+e^{-(\beta_0 + \beta_1 x)}}$$



Challenge:

How can you find the **best fit sigmoid curve?**

How to find the combination of β0 and β1 which fits the data best.

Introduction to Logistic Regression

Point no.	1	2	3	4	5	6	7	8	9	10
Diabetes	no	no	no	yes	no	yes	yes	yes	yes	yes

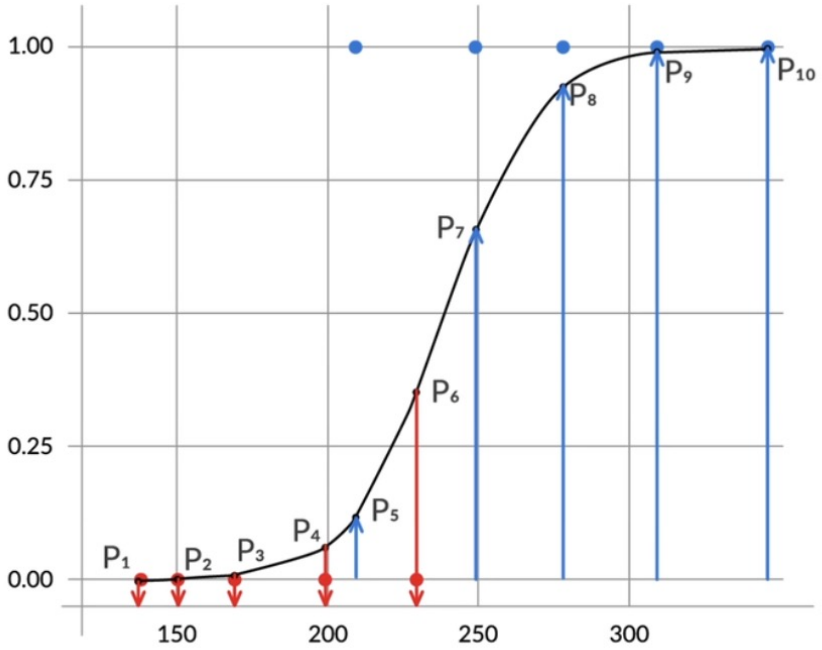
Cost Function?

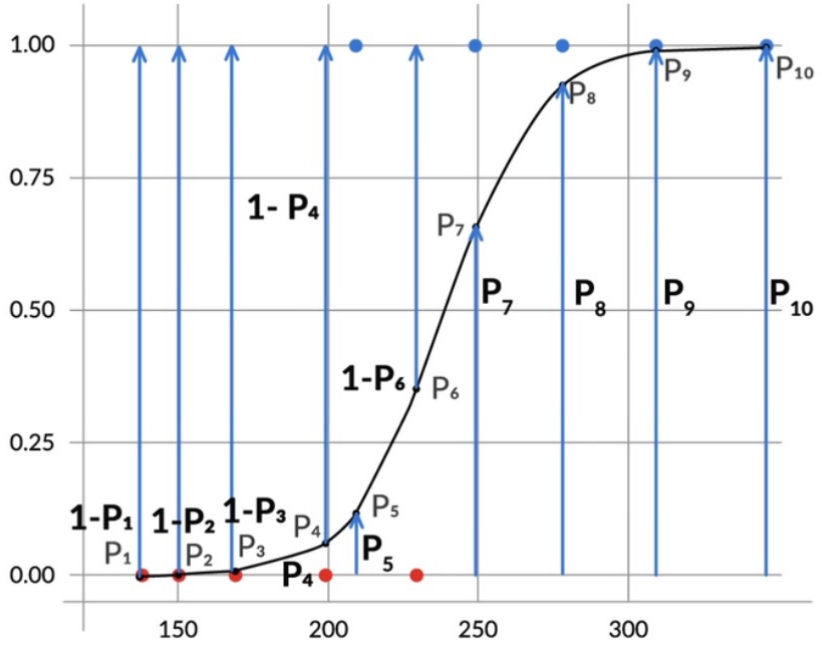
The best fitting combination of β0 and β1 will be the one which maximises the product:

(1−P1)(1−P2)(1−P3)(1−P4)(1−P6)(P5)(P7)(P8)(P9)(P10)

Maximum Likelihood Function

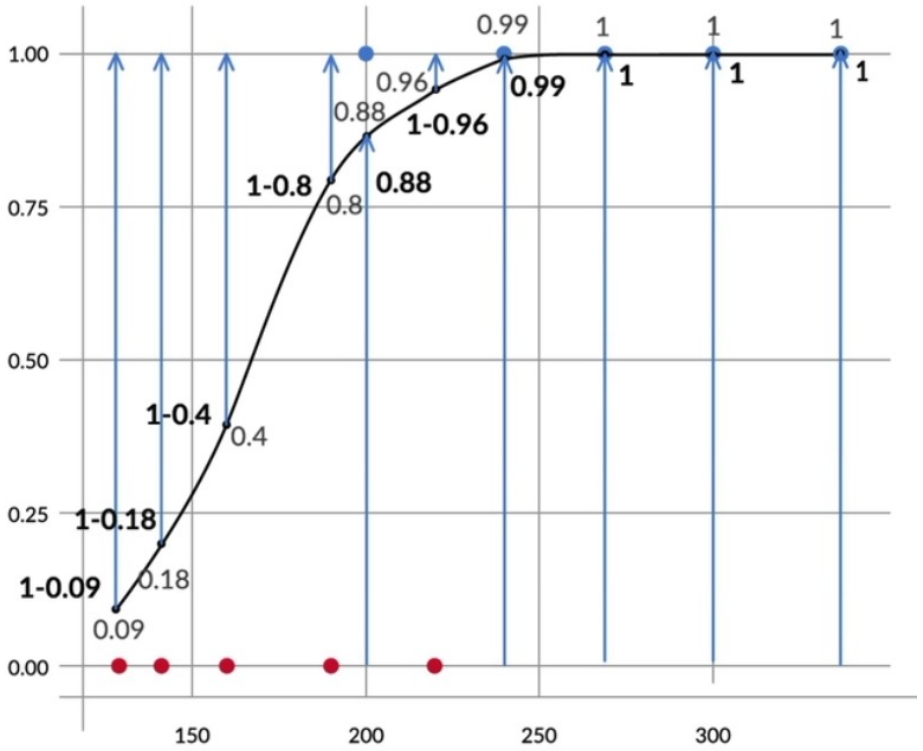
[(1−Pi)(1−Pi)------ for all non-diabetics --------]
** [(Pi)(Pi) -------- for all diabetics -------]*





Introduction to Logistic Regression

LIKELIHOOD FOR $\beta_0 = -10$ AND $\beta_1 = 0.06$



Likelihood = (1-0.09)(1-0.18)(1-0.4)(1-0.8)(1-0.96)
(0.88)(0.99)(1)(1)(1)

= 0.003

Minimizing the Cost with Gradient Descent

Gradient descent is an iterative optimization algorithm, which finds the minimum of a cost function.

In this process, it tries different values starting from a random combination and update them to reach the optimal ones, **minimizing the output**.

The update rule is the same as the one derived by using the sum of the squared errors (MSE) in linear regression. As a result, the same gradient descent formula is used for logistic regression as well.

By iterating over the training samples until convergence, it reaches the optimal parameters leading to minimum cost.

Odds and Log-Odds

Equation for logistic regression:

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

Note: The relationship between P and x is so complex that it is difficult to understand what kind of trend exists between the two. If you increase x by regular intervals of, say, 10, how will that affect the probability? Will it also increase by some regular interval? If not, what will happen?

LOGISTIC REGRESSION EQUATION

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

Where $\beta_0 = -13.5$ and $\beta_1 = 0.06$

Linearising Sigmoid Equation:

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

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

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

-13.5/0.06

Sugar Level	Probability
120	0.18%
130	9.96%
140	16.78%
150	26.88%
160	40.13%
170	54.99%
180	69.01%
190	80.23%
200	88.09%
210	93.10%

Logistic Regression using Python

In python, logistic regression can be implemented using libraries such as SKLearn and statsmodels, though looking at the coefficients and the model summary is easier using statsmodels.

Python Demo

Logistic Regression using Python

ROC Curve
AUC
Confusion Matrix
Accuracy
Precision
Recall
Sensitivity
Specificity
Finding Optimum Probability

Classification Errors

When making a prediction for a binary or two-class classification problem, there are two types of errors that we could make.

- **False Positive**. Predict an event when there was no event.
- **False Negative**. Predict no event when in fact there was an event.

By predicting probabilities and calibrating a threshold, a balance of these two concerns can be chosen by the operator of the model.

For example, in a smog prediction system, we may be far more concerned with having low false negatives than low false positives. A false negative would mean not warning about a smog day when in fact it is a high smog day, leading to health issues in the public that are unable to take precautions. A false positive means the public would take precautionary measures when they didn't need to.

Some Metrices

True Positive Rate = True Positives / (True Positives + False Negatives)

Sensitivity = True Positives / (True Positives + False Negatives)

False Positive Rate = False Positives / (False Positives + True Negatives)

Specificity = True Negatives / (True Negatives + False Positives)

False Positive Rate = 1 – Specificity

Positive Predictive Power = True Positives / (True Positives + False Positives)

Precision = True Positives / (True Positives + False Positives)

Recall = True Positives / (True Positives + False Negatives)

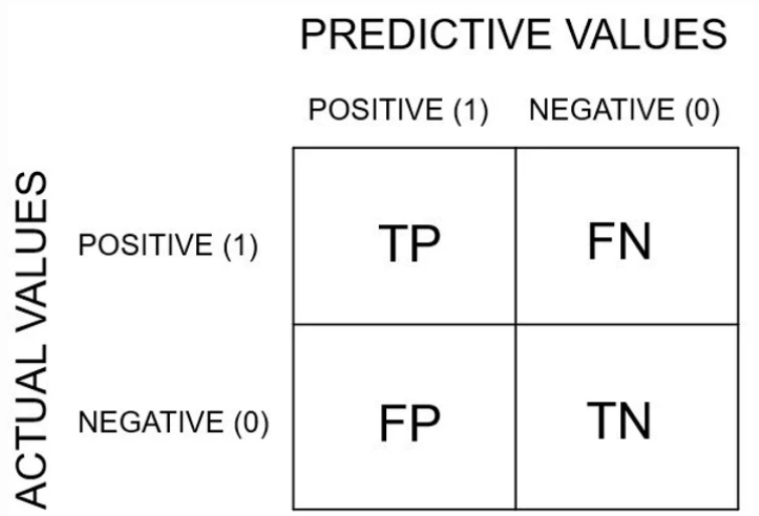
Sensitivity = True Positives / (True Positives + False Negatives)

Recall == Sensitivity

F1 Score: that calculates the harmonic mean of the precision and recall (harmonic mean because the precision and recall are rates).

$$\text{F1 Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$

Confusion Matrix



Some Metrices

The **True Positive Rate (Sensitivity)** is calculated as the number of true positives divided by the sum of the number of true positives and the number of false negatives. It describes how good the model is at predicting the positive class when the actual outcome is positive.

The **False Positive Rate (1- Specificity)** is calculated as the number of false positives divided by the sum of the number of false positives and the number of true negatives.

The False Positive Rate is also referred to as the **Inverted Specificity** where specificity is the total number of true negatives divided by the sum of the number of true negatives and false positives.

Precision is a ratio of the number of true positives divided by the sum of the true positives and false positives. It describes how good a model is at predicting the positive class. Precision is referred to as the positive predictive value.

Recall is calculated as the ratio of the number of true positives divided by the sum of the true positives and the false negatives. Recall is the same as **sensitivity**.

AUC-ROC Curve

AUC - ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1. By analogy, the Higher the AUC, the better the model is at distinguishing between patients with the disease and no disease.

The ROC curve is plotted with TPR against the FPR where TPR is on the y-axis and FPR is on the x-axis.

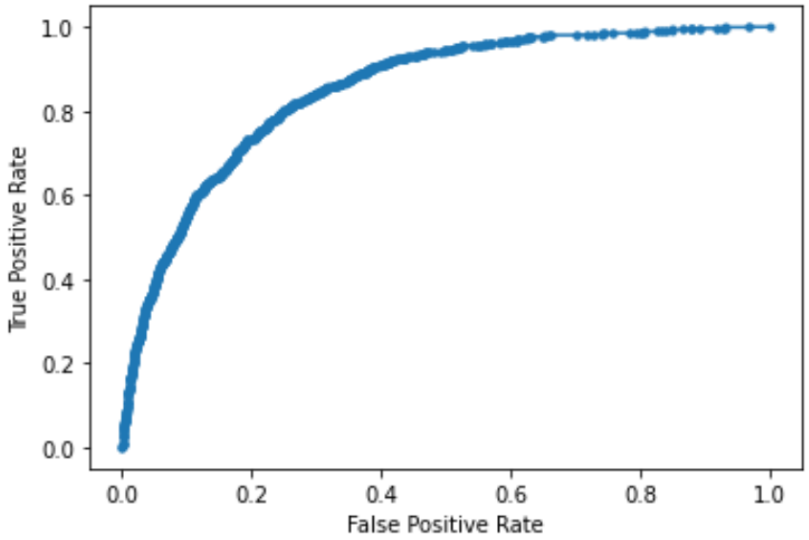
$$\text{TPR /Recall / Sensitivity} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

$$\text{FPR} = 1 - \text{Specificity}$$

$$= \frac{FP}{TN + FP}$$

ROC Curves summarize the trade-off between the true positive rate and false positive rate for a predictive model using different probability thresholds.



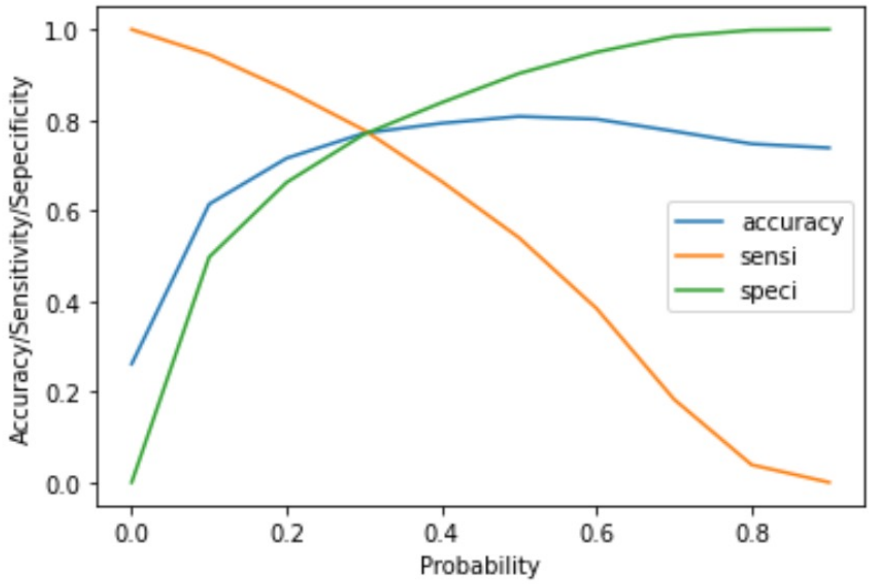
- A great model has AUC closer to 1
- A poor model has AUC closer to 0

Plot Accuracy-Sensitivity-Specificity for various Probabilities

One of the Optimization challenges of a Classification problem is to determine the right Probability Threshold. This is done visually by plotting the three key Metrics – Acruracy, Sensitivity and Specificity against Probability Thresholds.

When the probability thresholds are very low, the sensitivity is very high and specificity is very low. Similarly, for larger probability thresholds, the sensitivity values are very low but the specificity values are very high. And at about 0.3, the three metrics seem to be almost equal with decent values and hence, we choose 0.3 as the optimal cut-off point. The following graph also showcases that at about 0.3, the three metrics intersect.

We could've chosen any other cut-off point as well based on which of these metrics you want to be high. If you want to capture the ‘Positives' better, we could have let go of a little accuracy and would've chosen an even lower cut-off and vice-versa. It is completely dependent on the situation we are in. In this case, we just chose the 'Optimal' cut-off point to give you a fair idea of how the thresholds should be chosen.



	prob	accuracy	sensi	speci
0.0	0.0	0.261479	1.000000	0.000000
0.1	0.1	0.614384	0.944833	0.497387
0.2	0.2	0.715766	0.866356	0.662448
0.3	0.3	0.771028	0.777778	0.768638
0.4	0.4	0.792970	0.664336	0.838514
0.5	0.5	0.808005	0.540016	0.902889
0.6	0.6	0.801910	0.383838	0.949931
0.7	0.7	0.775295	0.183372	0.984869
0.8	0.8	0.747460	0.038850	0.998349
0.9	0.9	0.738521	0.000000	1.000000

Precision-Recall Curve

Precision-Recall is an useful measure of success of prediction when the classes are imbalanced.

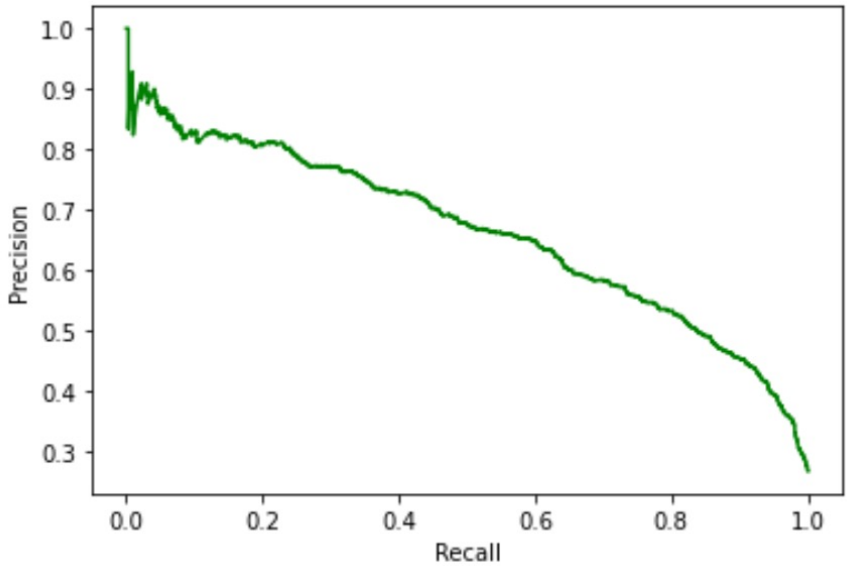
The **precision-recall curve** shows the trade-off between precision and recall for **different thresholds**.

- A high area under the curve represents both high recall and high precision
- High precision relates to a low false positive rate
- High recall relates to a low false negative rate
- High scores for both show that the classifier is returning accurate results (high precision), as well as returning a majority of all positive results (high recall)

A system with high recall but low precision will have most of its predicted labels are incorrect. A system with high precision but low recall is just the opposite, most of its predicted labels would be correct.

An ideal system with high precision and high recall will return many results, with all results labelled correctly.

Precision-Recall curves summarize the trade-off between the true positive rate and the positive predictive value for a predictive model using different probability thresholds.



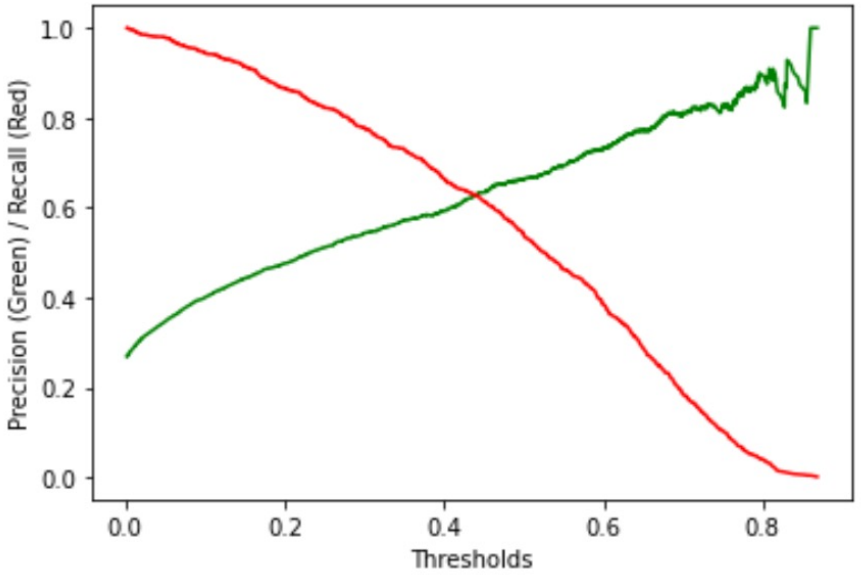
ROC curves are appropriate when the observations are balanced between each class, whereas precision-recall curves are appropriate for imbalanced datasets.

Precision-Recall vs Thresholds

Similar to the sensitivity-specificity trade-off, we see that there is a trade-off between precision and recall against thresholds.

As you can see, the curve is similar to what we got for sensitivity and specificity. Except now, the curve for precision is quite jumpy towards the end. This is because the denominator of precision, i.e. (TP+FP) is not constant as these are the predicted values of 1s. And because the predicted values can swing wildly, you get a very jumpy curve.

NOTE: This curve is useful when you would want to determine the Threshold for class prediction based on Precision and Recall values.



	Precision	Recall	thesholds
0	0.267568	1.000000	0.002434
1	0.267415	0.999223	0.002438
2	0.267471	0.999223	0.002438
3	0.267527	0.999223	0.002451
4	0.267582	0.999223	0.002471
5	0.267638	0.999223	0.002481
6	0.267694	0.999223	0.002487
7	0.267749	0.999223	0.002494
8	0.267805	0.999223	0.002508
9	0.267861	0.999223	0.002514

Logistic Regression Steps

To to summarise, the steps that you performed throughout model building and model evaluation were:

- Data cleaning and preparation
  - Combining three DataFrames
  - Handling categorical variables
    - Mapping categorical variables to integers
    - Dummy variable creation
  - Handling missing values
- Test-train split and scaling
- Model Building
  - Feature elimination based on correlations
  - Feature selection using RFE (Coarse Tuning)
  - Manual feature elimination (using p-values and VIFs)
- Model Evaluation
  - Accuracy
  - Sensitivity and Specificity
  - Optimal cut-off using ROC curve
  - Precision and Recall
- Predictions on the test set

Hope you have liked this Video.
Please help us by providing your Ratings and Comments for this Course!

Thank You!!
Manas Dasgupta

Happy Learning!!