Logistic Regression

Objectives

After completing this module, you should be able to:

Build Logistic Regression

Understand the concept of Maximum Likelihood Estimation

Decide Model Evaluation Parameters

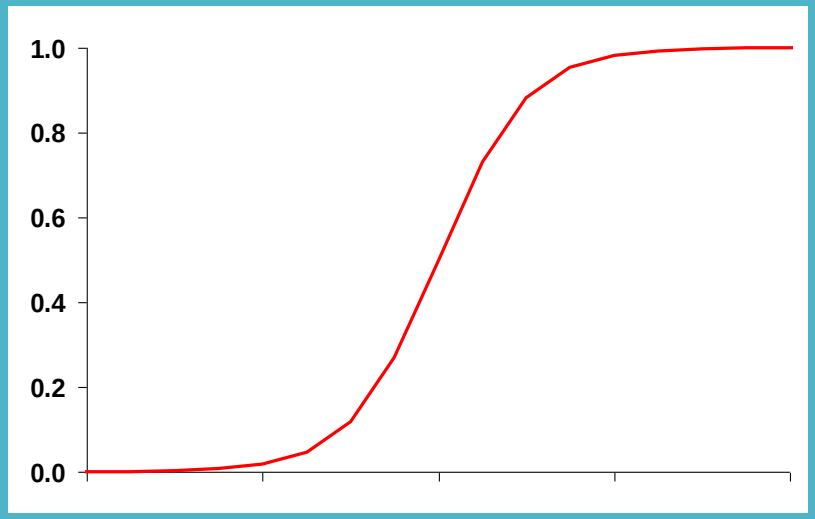
Regularization

WHAT IS LOGISTIC REGRESSION?

- Logistic Regression is a classification algorithm that models the probability of the output class.

- It estimates relationship between a dependent variable (target/label) and one or more independent variable (predictors) where dependent variable is categorical.

$$P(y|x) = \frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}}$$



Logistic Regression Features

- Expects a "smooth" linear relationship with predictors.
- Logistic Regression is concerned with probability of a discrete outcome.
- Slightly less prone to over-fitting
- Because fits a shape, might work better when less data available.

4

Assumptions

* Binary logistic regression requires the dependent variable to be binary.
* For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome.
* Only the meaningful variables should be included.
* The independent variables should be independent of each other. That is, the model should have little or no multicollinearity.
* The independent variables are linearly related to the log odds.
* Logistic regression requires quite large sample sizes.

5

Objective of Logistic Regression

Binary Classification:

- Given the subject and the email text predicting, Email Spam or not.
- Sunny or rainy day prediction, using the weather information.
- Based on the bank customer history, Predicting whether to give the loan or not.

Multi-Classification:

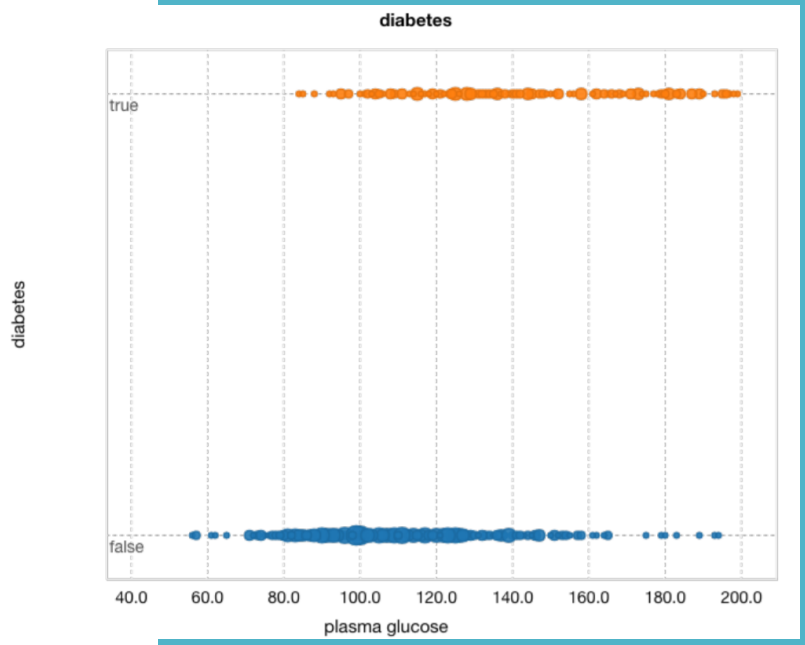
- Given the dimensional information of the object, Identifying the shape of the object.
- Identifying the different kinds of vehicles.
- Based on the color intensities, Predicting the color type

6

Logistic Regression – Diabetes Dataset

Y axis
True: Perosn has diabetes
Flase: No diabetes

X Axis
Feature – plasma glucose

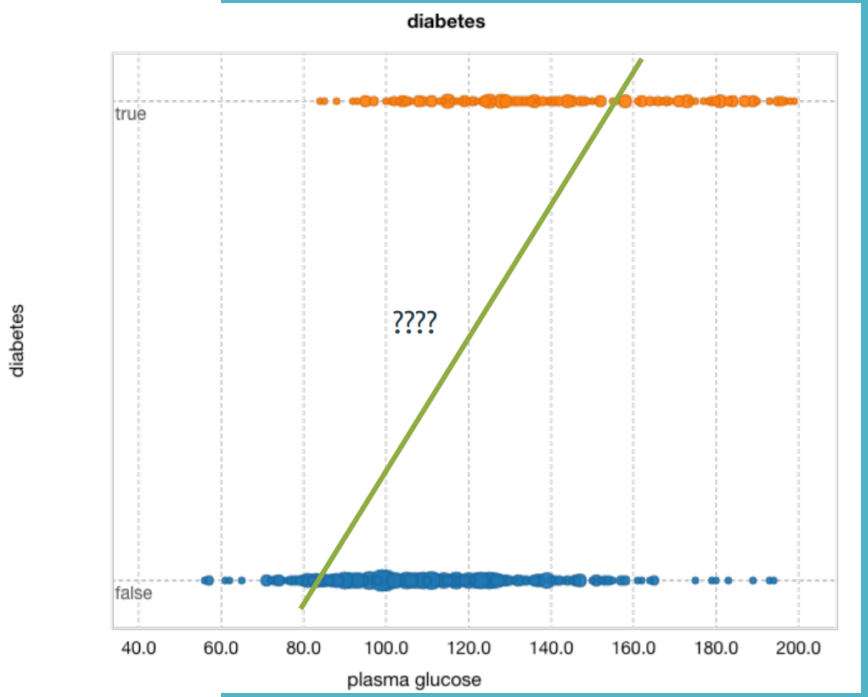


Logistic Regression – Diabetes Dataset

$$\hat{y} = mx + c$$

$\hat{y} = Value\ predicted\ by\ current\ Algorithm$

Linear Regression in one Variable



Logistic Regression – Diabetes Dataset

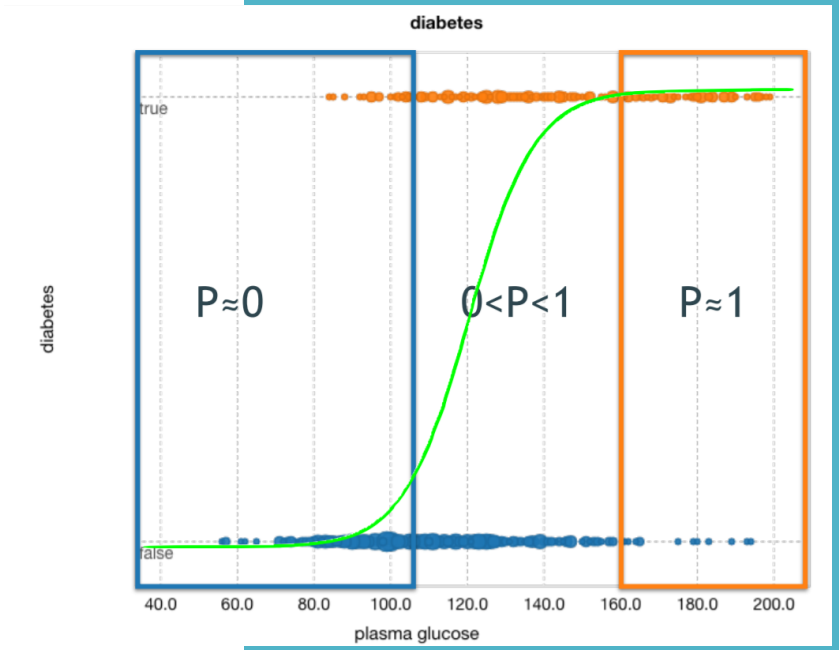
$$\hat{y} = mx + c$$

$$p = \frac{1}{1 + e^{-\hat{y}}}$$

Logistic Regression – Diabetes Dataset

$$p = \frac{1}{1 + e^{-\hat{y}}}$$

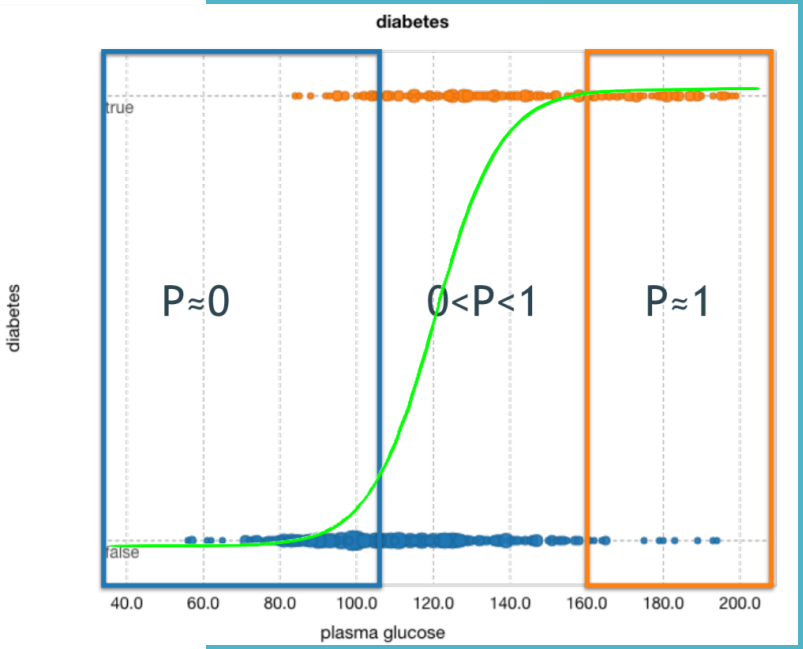
$$p = \frac{1}{1 + e^{-(mx+c)}}$$



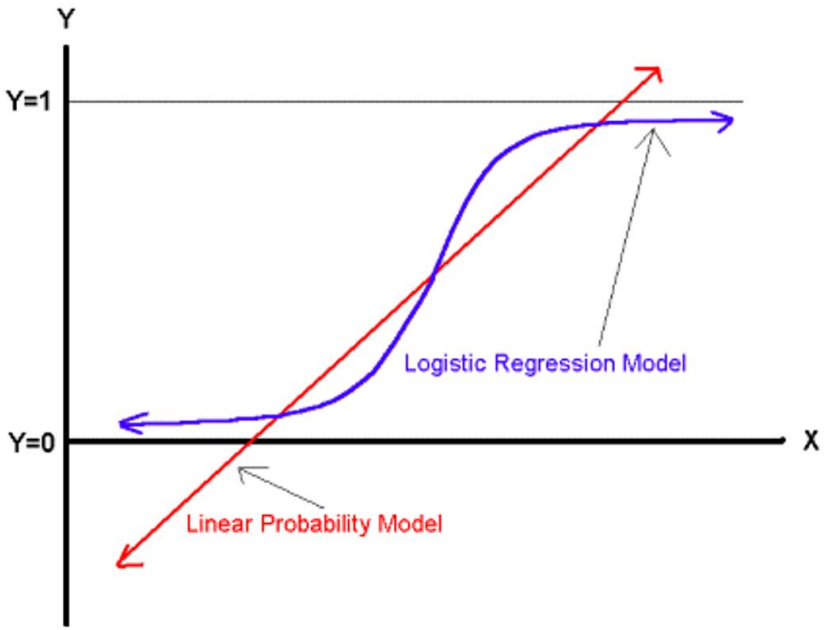
Logistic Regression – Diabetes Dataset

$$p = \frac{1}{1 + e^{-(mx+c)}}$$

$$ln\left(\frac{p}{1-p}\right) = mx + c$$



Comparing Linear Probability Model and Logistic Regression Model



Binary Classification using Logistic Regression

$$Sigmoidal\ Function$$

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

Multinomial Classification using Logistic Regression

$$Softmax\ function$$

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}\ for\ j = 1,2, \ldots . . K$$

13

Logistic Regression Performance Analysis

Confusion Matrix

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.

	1 (Predicted)	0 (Predicted)
1 (Actual)	True Positive	False Negative
0 (Actual)	False Positive	True Negative

Confusion Matrix

True positive = correctly identified

False positive = incorrectly identified

True negative = correctly rejected

False negative = incorrectly rejected

Accuracy

It determines the overall predicted accuracy of the model. It is calculated as

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

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP} + \boldsymbol{TN}}{\boldsymbol{TP} + \boldsymbol{TN} + \boldsymbol{FP} + \boldsymbol{FN}}$$

Recall or True Positive Rate (TPR) –

The recall is intuitively the ability of the classifier to find all the positive samples.

It indicates how many positive values, out of all the positive values, have been correctly predicted. The formula to calculate the true positive rate is (TP/TP + FN).

Also, TPR = 1 - False Negative Rate. It is also known as Sensitivity or Recall. Recall (TPR)=TP/(TP+FN)

False Positive Rate (FPR)

It indicates how many negative values, out of all the negative values, have been incorrectly predicted. The formula to calculate the false positive rate is (FP/FP + TN).

Also, FPR = 1 - True Negative Rate.

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

True Negative Rate (TNR)

It indicates how many negative values, out of all the negative values, have been correctly predicted. The formula to calculate the true negative rate is (TN/TN + FP).

It is also known as Specificity.

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

False Negative Rate (FNR)

It indicates how many positive values, out of all the positive values, have been incorrectly predicted.

The formula to calculate false negative rate is (FN/FN + TP).

$$\boldsymbol{FNR = \frac{FN}{FN+TP}}$$

Precision

Precision: It indicates how many values, out of all the predicted positive values, are actually positive. It is formulated as:(TP / TP + FP).

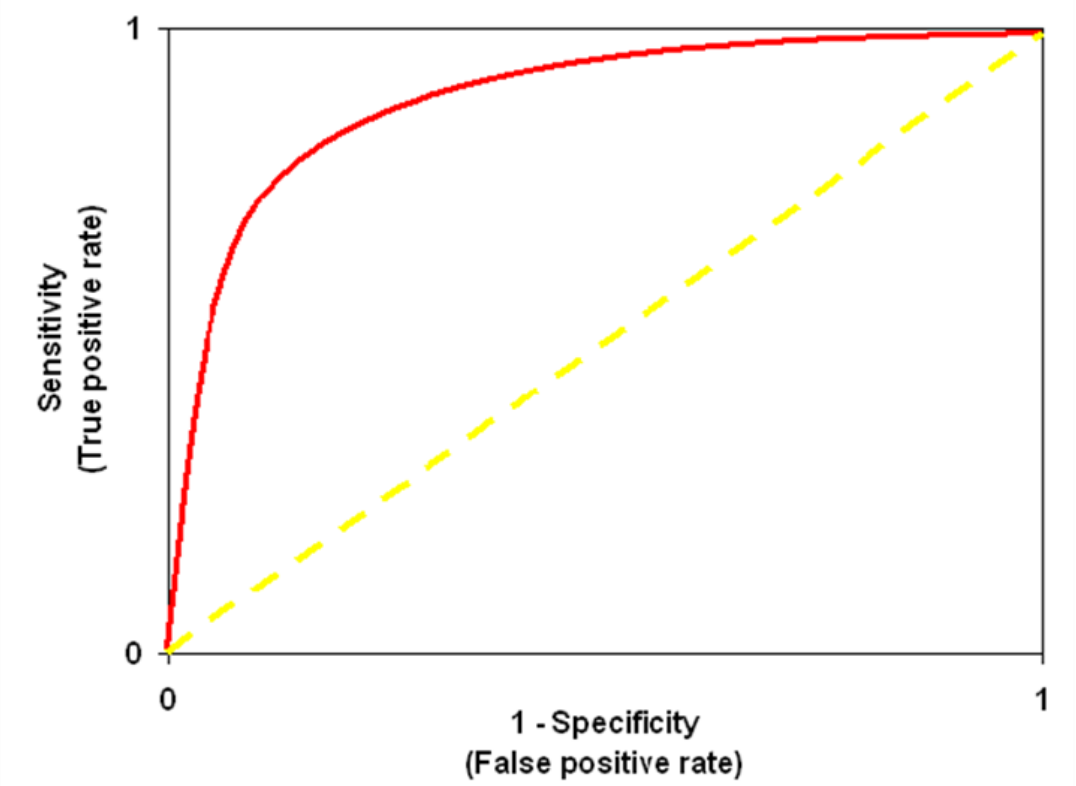
$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}}$$

F1 Score

F1 Score: F1 score is the harmonic mean of precision and recall. It lies between 0 and 1. Higher the value, better the model. It is formulated as 2((precision*recall) / (precision+recall)).

$$\boldsymbol{F1\ Score = 2 * \frac{(precision * recall)}{(precision + recall)}}$$

Receiver Operator Characteristic (ROC)

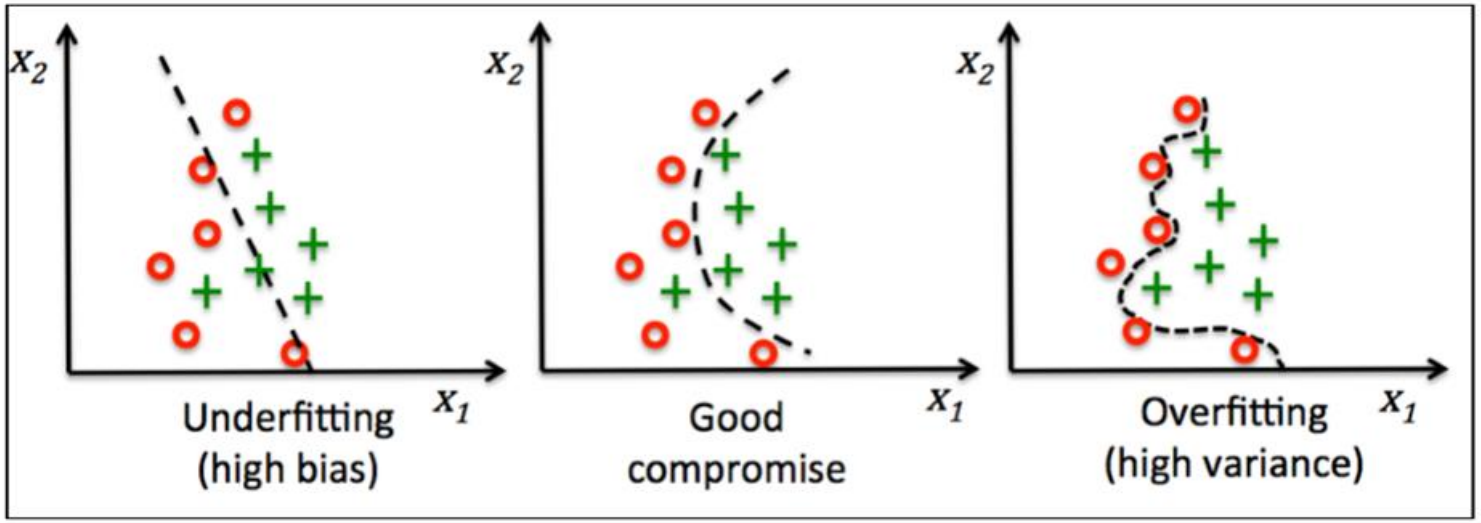


Overfitting & Generalization

Overfitting is a common problem in machine learning, where a model performs well on training data but does not generalize well to unseen data (test data).

Overfitting occurs when a model is excessively complex, such as having too many parameters relative to the number of observations.

Overfitting & Generalization



Overfitting & Generalization

How to avoid overfitting?

One of the ways to combat over-fitting is to increase the training data size.

Another way to combat over-fitting is to perform early stopping.

Maximum Likelihood Estimation (MLE)

MLE is a statistical method for estimating the coefficients of a model.

The likelihood function (L) measures the probability of observing the particular set of dependent variable values (p1, p2, ..., pn) that occur in the sample:

$$L = \text{Prob}(p1 * p2 * * * pn)$$

The higher the L, the higher the probability of observing the ps in the sample.

Maximum Likelihood Estimation (MLE)

MLE involves finding the coefficients (□, □) that makes the log of the likelihood function (LL < 0) as large as possible

Or, finds the coefficients that make -2 times the log of the likelihood function (-2LL) as small as possible

The maximum likelihood estimates solve the following condition:

$$\{Y - p(Y=1)\}Xi = 0$$

summed over all observations, i = 1,…,n

Summary

This module covered the following topics:

Logistic Regression

Maximum Likelihood Analysis

Model Evaluation

Regularization

Thank you