Logistic Regression

Logistic Regression

In layman's terms, classification in the process of taking a collection of data and organizing it into groups or categories based on some common characteristics.

For example, if you have a dataset of images, a classification algorithm might be used to automatically organize the images into groups based on the objects or scenes they depict.

This can be done using a training dataset that includes labeled examples of the different classes or categories.

The algorithm uses this training data to learn the characteristics that distinguish each class, and then uses this knowledge to make predictions on new, unbalanced data.

In more detailed terms, classification is a type of supervised learning algorithm, which means that it is trained on a dataset that includes both the input data and the corresponding labels.

The training dataset is used to teach the algorithm about the relationship between the input data and the labels.

Classification

This can be done using a variety of techniques, such as decision trees, support vector machines, and neural network. Once the algorithm has been trained. It can be used to make predictions on new, unseen data.

For example, if you have a dataset of images, the classification algorithm might take an image as image as input and predict which class or category it belongs to such as 'cat' or 'dog'.

The algorithm's ability to make accurate predictions can be evaluated using a separate test dataset, which includes input data and known labels. This allows you to measure the performance of the algorithm and make any necessary adjustments to improve its accuracy.

applications

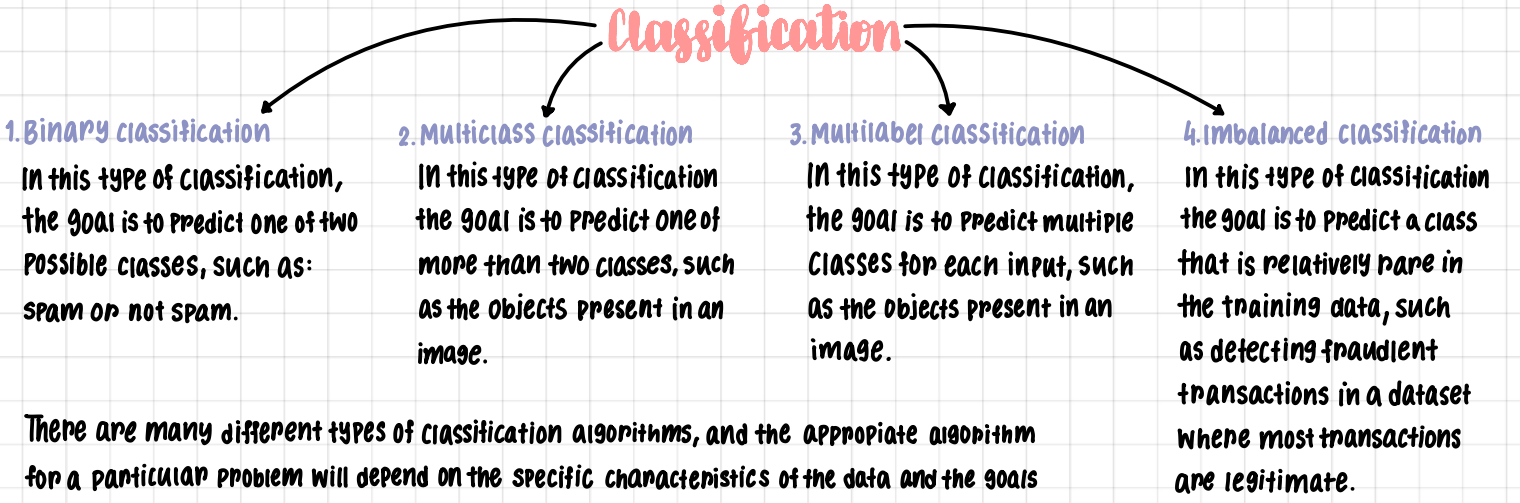
Some basic examples of classification problems in machine learning include:

Classification problems	In this problem, the goal is to classify
1. Spam detection	an email as either spam or not spam.
2. Sentiment analysis	a piece of text as expressing positive.
3. Fraud detection	transactions as either fraudulent or legitimate.
4. Medical diagnosis	a patient's medical condition based on their symptoms and test results.
5. Image classifications	an image as containing a particular object or not, such as a dog or a cat.

These are just a few examples of classification problems that can be tackled using machine learning. Other examples include language translation, speech recognition, and many more.

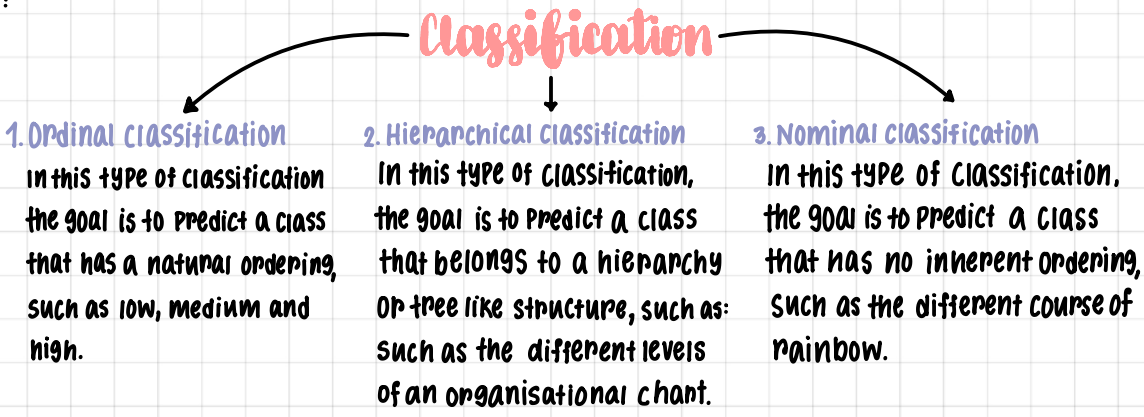
types of Classification

There are several types of classification data, including:



There are many different types of classification algorithms, and the appropiate algorithm for a particular problem will depend on the specific characteristics of the data and the goals of the model. Some common types of classification algorithms include:

In addition to the types of the classification data mentioned in the previous answer, there are few other types that are worth mentioning. These include:



Again, the appropiate type of classification data for a particular problem will depend on the specific characteristics of the data and the goals of the machine learning model.

Problem Statement

The problem of predicting whether an individual will default on their credit card payment, on the basis of their annual income and monthly credit card balance, is a supervised learning problem in the field of machine learning.

In this problem, the goal is to build a model that can take in an individual's annual income and monthly credit card balance as input, and predict whether or not they will default on their credit card payment.

The model will be trained on a labeled dataset that contains examples of individuals and whether or not they defaulted on their credit card payment. This labeled training data will be used to learn the relationship between an individual's annual income and monthly credit card balance, and their likelihood of defaulting on their payment.

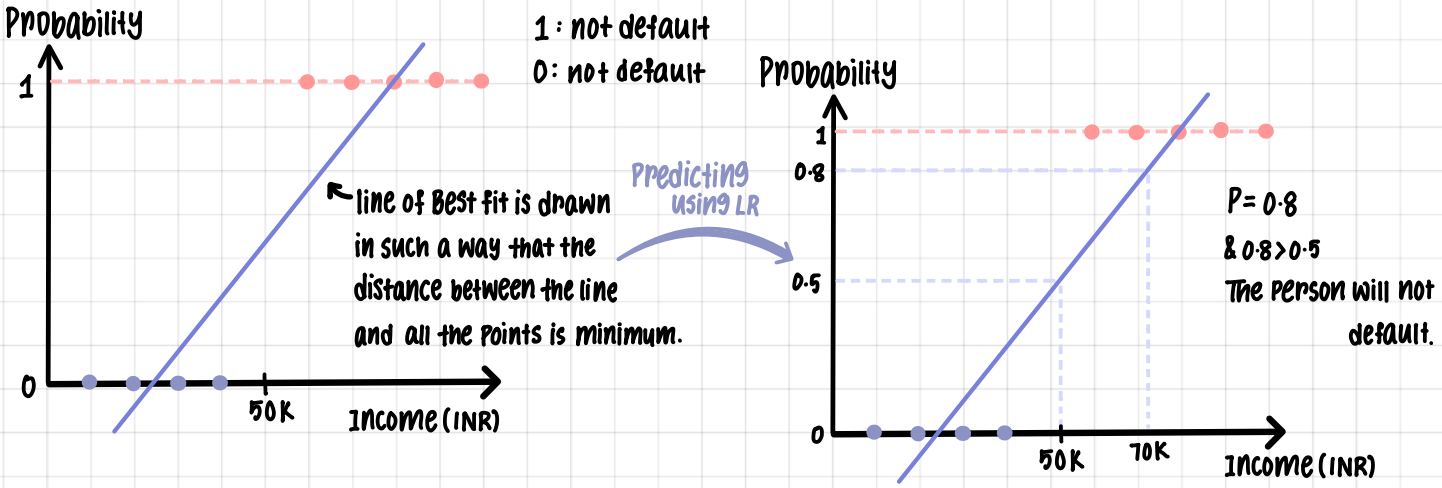
Once the model is trained, it can be used to make predictions on new, unseen data. This could be used, for example, by a credit card company to assess the risk of a potential customer defaulting on their payment, and to make decisions about whether to approve or deny their credit card application.

Overall, this problem involves using machine learning to build a model that can predict an individual's likelihood of defaulting on their credit card payment, based on their annual income and monthly credit card balance.

The individuals who defaulted in a given month are shown in orange, and those who did not in blue.
It appears that individuals who defaulted tended to have higher credit card balances than those who did not.
We learn how to build a model to predict default (Y) for any given value of balance (x1) and income x2.

Why not Linear Regression

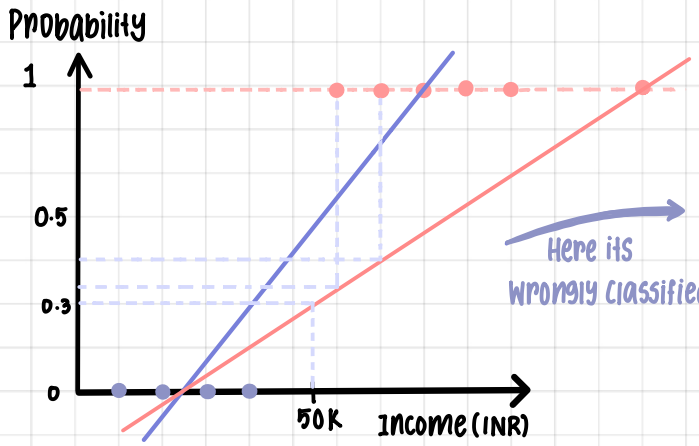
You have a problem to solve which is predicting whether a person will default or not given its income.

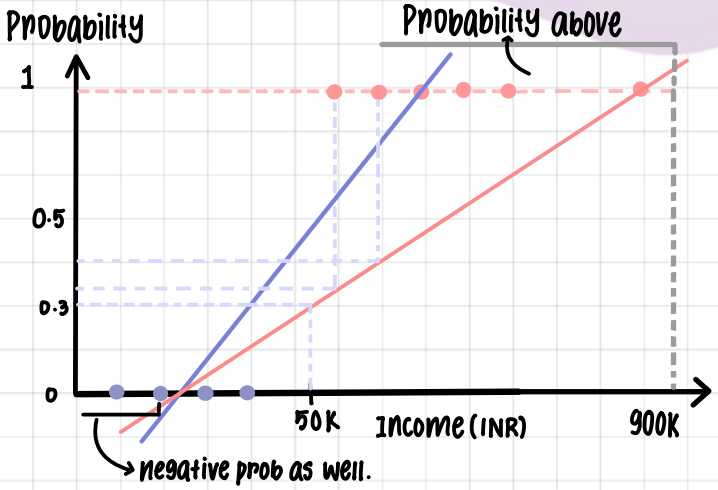


∴ If Y ≥ 0.5, then the person is not default.
If Y < 0.5, then the person is default.

Then where's the problem → linear regression is highly affected by inclusion of outliers.

1 2 3 4 5 6 7 1000 180
(1000, 180 → outliers)





yes, the person defaulted → yes

DEFAULT

no → No, the person haven't defaulted

→ Modelling the probability that Y belongs to any of the category

$Pr(default = Yes | Balance) \longrightarrow$ [0-1] output values

E.g. P(balance) > 0.5 ; then the prediction would be yes!

Threshold → Changeable!!

SO, HOW should we model the relationship between $p(x) = Pr(Y=1|x)$ and X?

$p(x) = \beta_0 + \beta_1 X$ → like linear hypothesis problem with this approach:

→ for balances close to 0, we can predict the "-" probability.

→ Probability can be greater than 1.

It must fall between 0 & 1 ← The pred. are not sensible, since of course than the prob. of default, regardless of credit card balance.

If use straight line:

→ p(x) < 0 (for some values of x)

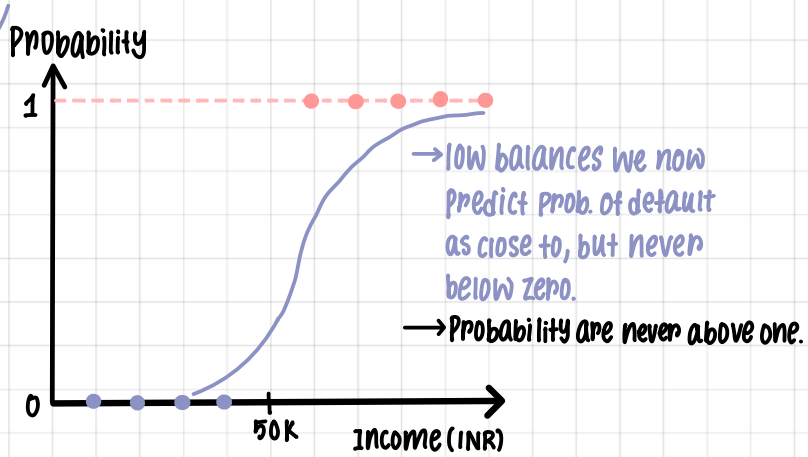
→ p(x) > 0 (for some values of x)

To fix this problem → we must model p(x) using a function that gives outputs between 0 and 1 for all values of x.

logistic function

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

→ log odds



E9: Recommended the service:

A survey describes a survey of 250 customers of an automobile dealership. The customers here asusual if they would recommend the service department to a friend. The no of who recommended Yes was 210.

$n = 250$; $X = 210$; $\hat{p} = \frac{210}{250} = 0.84$ (proportion) $\xrightarrow{\text{log. reg.}}$ works with odds rather than proportions.

Odds are simply the ratio of the probability for the two possible outcomes.

$$\text{Odds} = \frac{\hat{p}}{1-\hat{p}} \; ; \; \hat{p} = 0.84 \; ; \; 1-\hat{p} = 1-0.84 = 0.16$$

$$= \frac{0.84}{0.16} = 5.125 \xrightarrow{\text{round}} 5 = \frac{5}{1} \; ;$$

5:1 → five times higher probability of a customer recommending the deruice department than not recommending it.

Sample proportion of women who are insta users is given as 61.08% and proportion of man 43.98%

$x = \begin{cases} 1 \rightarrow \text{if the person - woman} \\ 0 \rightarrow \text{if the person - man} \end{cases}$

$\text{Odds} = \frac{\hat{p}}{1-\hat{p}} = \frac{0.6108}{1-0.6108} = 1.5694$ (for women)

$\text{odds} = \frac{\hat{p}}{1-\hat{p}} = \frac{0.4398}{1-0.4398} = 0.7851$ (for man)

Logistic Regression

The logistic function is defined as:

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

- pred. prob that Pred will be 1
- regression coefficient estimated from data
- Euler's number

Derivation:

→ Derived from Bernoulli distribution, probability distribution that models the outcome of a single trial that can result in either True/"sucess" or "faliure".

Let P(x) be the probability observing success outcome (Y=1) given X.

Then the probability of observing a 'faliure' outcome (Y=0) is 1 - P(x)

Odds of observing success → ratio of $\frac{P(x)}{1-P(x)}$ $\quad \text{Odds}(Y=1|X) = \frac{P(x)}{1-P(x)}$

Odds can take any '+' values from 0 to ∞.

↳ transformed to log odds (logit).

This takes from neg infinity to '+' ∞.

$\text{logit}(P(x)) = \log\left(\frac{P(x)}{1-P(x)}\right)$ take the inverse of the logit transformation, maps log-odds to the prob

$P(x) = \frac{1}{1+\exp^{(-\text{logit}+P(x))}}$ Substituting logit yeilds

$\text{logit}(P(x)) = \log\left(\frac{P(x)}{1-P(x)}\right)$

$P(x) = \frac{1}{1+\exp(-[\beta_0+\beta_1 x])}$

↓

equation simplified by multiplying both numerator & denominator by $\exp(\beta_0+\beta_1 x)$

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

So, what we generally use?

What function we generally use?

The two forms of the logistic function, $P(X) = e^{\wedge}(\beta_0+\beta_1 x)/(1+e^{\wedge}(\beta_0+\beta_1 x))$ and $P(x) = 1/[1+\exp(-\beta_0-\beta_1 x)$, are mathematically equivalent and produce identical results.

Both forms map the linear combination of predictor variables and corresponding regression coefficients to the range of [0,1], which ensures that the predicted values are valid probabilities.

The choice between the two forms of the logistic function is largely a matter of convention and personal preference.

Some people find the first form, which involves the exponential function, to be more intuitive and easier to work with, while others prefer the second form, which involves the negative exponential function.

In practice, most software packages that implement logistic regression use the first form of the logistic function, $p(x) = e^{(\beta_0+\beta_1 x)} / (1 + e^{(\beta_0+\beta_1 x)})$, because it has some computational advantages.

Specially, it avoids the need to compute the negative exponentials, which can be computationally expensive and may introduce numerical stability issues in some cases.

Worked Example:

Suppose we have a dataset of 100 customers who have applied for a loan, and we want to model the probability of a customer defaulting on the loan (ie $Y=1$) as a function of their credit score (ie, X). The credit scores range from 500 to 800, and we have the following logistic regression model:

$$\text{logit } P(X) = \beta_0 + \beta_1 X$$

Where $P(X)$ is the predicted probability of default given the credit score X,
β_0 is the intercept
β_1 is the slope

Let's say that we have estimated the following coefficients from the data using maximum likelihood estimation:

$$\beta_0 = -0.3, \ \beta_1 = 0.01$$

Given these coefficients, we can calculate the predicted probability of default for a customer with a credit score of 700 using the following steps:

Calculate the logit of the predicted probability:
$\text{logit } P(x) = \beta_0 + \beta_1 x = -0.3 + 0.01 \times 700 = 3.0$

Apply the logistic function to the logit:
$P(X) = 1 / [1 + \exp(-\text{logit}(P(x))] = 1 / [1 + \exp(-3.0)] = 0.0498$

Therefore, the predicted probability of default for a customer with a credit score of 700 is 0.0498 or 4.98%. This means that we estimate that there is a 4.98% chance that the customer will default on the loan given their credit score of 700.

Estimating the Coefficient in logistic Regression

Introduction: In logistic regression, we use the logistic hypothesis function to model the relationship between the binary response variable and the predictor variables.

Estimating the coefficients: To use the logistic hypothesis function, we need to estimate the coefficients, which are the values of β_0 and β_1 in the logistic function.

Maximum likelihood Estimation: The most common method for estimating the coefficients is maximum likelihood estimation, which finds the values of β_0 and β_1 that maximise the likelihood of observing the data given the logistic hypothesis function.

Example: Let's say we have a dataset of 100 customers who have applied for a loan, and we want to model the probability of a customer defaulting on the loan as a function of their credit score. The logistic hypothesis function is:

$$\text{logit} = \beta_0 + \beta_1 x$$

where $P(x)$ is the predicted probability of default given the credit score X, β_0 is the intercept and β_1 is the slope.

Coefficients: The coefficients β_0 and β_1 represent the intercept and the slope of the logistic regression model, respectively. They describe the relationship between the log-odds of the probability of default and the credit score.

Interpretation: For example, if β_1 is positive, it means that as the credit score increases, the log-odds of defaulting on the loan increase. If β_0 is negative, it means that the baseline log-odds of defaulting on the loan are lower, even for a credit score of zero.

Conclusion: By estimating the coefficients, we can use the logistic hypothesis function to make predictions of the probability of default given the credit score, and gain valuable insights to the relationship between the binary response variable and predictor variable.

Estimating regression coefficients

- We used the least square approach to estimate the coefficients of linear regression.
- We could use least squares to fit the logistic model, but generally we use general method of maximum likelihood.

We will learn about the difference between least squares and maximum likelihood estimation.

Intvision behind this:

We seek to estimates for β_0 and β_1 → S.t. the Predicted Prob $\hat{p}(x_i)$ of default for each individual. → corresponds as closely as possible to the individual default status.

Find β_0 & β_1 → Plugs in $P(x)$ which yeild

- a no. close to one for all individuals who defaulted.
- a no. close to "0" for individuals who don't

Maximum likelihood Method

The maximum likelihood method is a stastical technique for estimating the parameters of a statistical model based on the observed data. The goal of the maximum likelihood method is to find the values of the parameters that maximize the probability of the observed data given the model.

The specific form of the likelihood function depends on the form of the model and the distribution of the data. For example, in logistic regression, the model is based on the logistic function and the data are assumed to be independently and identically distributed according to a Bernoulli distribution. In this case, the likelihood function has the form:

$$L(\beta) = \prod_{i:y_i=1} P(x_i) \prod_{i':y_i'=0} (1-P(x_i'))$$

$$(1 - P(y_i | x_i', \beta)^{(1-y_i)}$$

$$P(y_i|x_i, \beta)^{y_i}$$

↓ predicted probability

→ likelihood function

Suppose we are interested in predicting the probability of a customer defaulting on a credit card given their account balance and credit score. We have a dataset containing the account balance, credit scores, and default status (0 = not defaulted, 1 = defaulted) for a sample of customers.

We can use logistic regression to model the relationship between the account balance, credit score, and the probability of default. In this case, the logistic regression model would have the form:

Discrete Probability Distribution

Bernoulli distribution:

Suppose you have:

- ⟶ an experiment
- ⟶ expt. results in one of the two possible outcomes, success or failure or binary response.
- ⟶ P(success) = p and P(failure) = 1-p

Let Y=1 if a success occurs and Y=0 if a failure occurs.

Then X has a Bernoulli Distribution

$P(Y=y) = p^y(1-p)^{1-y}$ } known as the pmf.

for $y = 0, 1$

$P(Y=1) = p^1(1-p)^{1-1} = p$

$P(Y=0) = p^0(1-p)^{1-0} = 1-p$

Maximum likelihood Estimation

⟶ Helps to estimate βs.

Why??

Labels are binary and pred. will be either one of them 0 or 1 which means:

$Y \sim Ber(p)$ where

$$p = \frac{e^{(z)}}{1+e^{(z)}} \text{ or } \frac{1}{1+e^{-z}}$$ ↳ sigmoid function

$$= \beta_0 + \beta_1 x$$

Pmf of Bernoulli

$$P(Y=y|X=x) = p^y(1-p)^{1-y}$$

$$= \left(\frac{1}{1+e^{-(\beta_0+\beta_1 x)}}\right)^y * \left(1 - \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}\right)^{(1-y)}$$

for any data point

We can write the likelihood of the all data point.

$$L(\beta) = \prod_{i=1}^{m} P(Y=y^{(i)} | X=x^{(i)})$$ "likelihood of independent training labels"

$$= \prod_{i=1}^{m} P(x_i) \prod_{i=1}^{m} (1-P(x_i))$$

$P(x_i)^{y_i}$ ↳ $P(y_i|x_i;\beta)^{y_i}$

$(1-P(x_i))^{(1-y_i)}$ ↳ $1-P(y_i|x_i;\beta)^{(1-y_i)}$

Worked Example

Suppose we have a dataset of 4 customers who have applied for a loan, and we want to model the probability of a customer defaulting on the loan (ie., Y=1) as a function of their credit score (i.e. X). The credit scores and binary response variables are given in the following table:

Customer	X	Y
1	600	1
2	550	0
3	700	1
4	650	0

Let's assume that we have estimated the following coefficients from the data using MLE:

$\beta_0 = -3.0 \quad \beta_1 = 0.01$

Given these coefficients, we can calculate the predicted probabilities of default for customer in the dataset:

For customer 1 with a credit score of 600, the predicted probability of default is:

$p(X) = 1 / [1 + \exp(-\text{logit}(p(X)))] = 1 / [1 + \exp(-(-3.0 + 0.01 * 600))] = 0.9525$

For customer 2 with a credit score of 550, the predicted probability of default is:

$p(X) = 1 / [1 + \exp(-\text{logit}(p(X)))] = 1 / [1 + \exp(-(-3.0 + 0.01 * 550))] = 0.0478$

For customer 3 with a credit score of 700, the predicted probability of default is:

$p(X) = 1 / [1 + \exp(-\text{logit}(p(X)))] = 1 / [1 + \exp(-(-3.0 + 0.01 * 700))] = 0.9975$

For customer 4 with a credit score of 650, the predicted probability of default is:

$p(X) = 1 / [1 + \exp(-\text{logit}(p(X)))] = 1 / [1 + \exp(-(-3.0 + 0.01 * 650))] = 0.7876$

Given the predicted probabilities of default, we can calculate the likelihood of observing the data for each customer:

For a customer 1 with a binary response variable Y=1, the likelihood of observing the data is:

$p(X)^Y * (1 - P(X))^{(1-Y)} = 0.9525^1 * (1 - 0.9525)^{(1-1)} = 0.9525$

For a customer 2 with a binary response variable Y=0, the likelihood of observing the data is:

$p(X)^Y * (1 - P(X))^{(1-Y)} = 0.0478^0 * (1 - 0.0478)^{(1-0)} = 0.9522$

For a customer 3 with a binary response variable Y=1, the likelihood of observing the data is:

$p(X)^Y * (1 - P(X))^{(1-Y)} = 0.9975^1 * (1 - 0.9975)^{(1-1)} = 0.9975$

For a customer 4 with a binary response variable Y=0, the likelihood of observing the data is:

$p(X)^Y * (1 - P(X))^{(1-Y)} = 0.7876^0 * (1 - 0.7876)^{(1-0)} = 0.2124$

$L(\beta_0, \beta_1) = \prod_{i=1}^{n} P(X_i)^{Y_i} * (1 - P(X_i))^{1-Y_i} = 0.9525 * 0.9522 * 0.9975 * 0.2124 = 0.192158\,93286$

The likelihood value of 0.1921 in this example represents the probability of observing the binary response variable Y for all 4 customers given the credit scores X and the estimated coefficients β_0 and β_1. A higher likelihood value indicates a better fit of the logistic regression model to the data set, as it means that the estimated coefficients are more likely to have generated the observed data.

A likelihood value of 0.1921 is not considered a good fit of the logistic regression model to the data, as it indicates a relatively low probability of observing the binary response variable Y given the credit scores X and estimated coefficients β_0 and β_1. A better fit of the model to the data would typically result in a higher likelihood value.

log likelihood

$$L(\beta_0, \beta_1) = \sum_{i=1}^{n} \left[y_i \log(p(x_i)) + (1-y_i) \log(1-p(x_i)) \right]$$

Why log likelihood?

Simplification of calculations

- Taking the logarithm of a product of probabilities transforms it into a sum of logarithms, which is easier to work with and less prone to numerical errors

Convenient optimization

- The log likelihood is a continuous and differentiable function, making it easier to optimize using gradient descent or other optimization algorithms.
- The objective is to find the values of β_0 and β_1 that maximise the log likelihood (equivalent to minimizing the negative log likelihood)

Additivity property

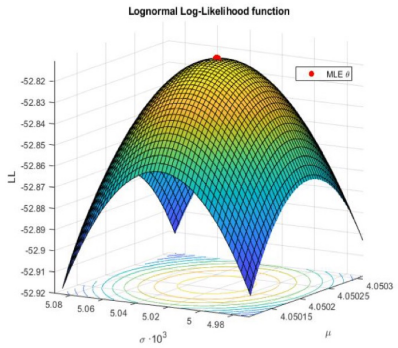
- Provides a convenient way to interpret the goodness of fit of the model of the data
- A higher log likelihood indicates a better fit of the model to the data, while a lower log likelihood indicates a poorer fit
- Can be used to compare different models, with the model with the highest log likelihood typically considered the best model.

Gradient Ascent

Gradient ascent is a method for finding the maximum of a function. It is used in logistic regression to find the best-fit to the data.
Imagine you are hiking up a mountain. The goal is to reach the highest point of the mountain, which is the peak. You start at the bottom of the mountain and need to find the best path to the peak.

Just like hiking up a mountain, gradient ascent finds the best path to the maximum of a function. The function represents the log likelihood in logistic regression, and the goal is to find the maximum log likelihood, which corresponds to the best-fit model.

Gradient ascent works by iteratively updating the values of the parameters in the direction of the steepest ascent of the log likelihood. The gradient of the log likelihood tells you the direction of steepest ascent. you move in that direction by a small step size at each iteration.



Think of the gradient as a compass. The gradient points you in the direction of the steepest ascent, just like a compass points you in the direction of north. You keep following the gradient until you reach the maximum log likelihood.

In logistic regression, gradient ascent is used to find the maximum log likelihood and the best-fit model to the data. The log likelihood is a measure of how well the model fits the data, and the goal is to find the values of the parameters that maximize the log likelihood. Gradient ascent helps us achieve this by iteratively updating the values of the parameters in the direction of the steepest ascent of the log likelihood.

So we apply gradient ASCENT algorithm in order to find our beta values which maximizes our log likelihood.

Gradient Ascent for learning &
Repeat {

$$\beta_j^{new} = \beta_j^{old} + \alpha \cdot \frac{\partial L(\beta)}{\partial \beta_j^{old}} \quad (\text{for } j=0,1,2,\ldots,n)$$

}

learning rate

partial derivative of likelihood

Because we want to maximise

$$L(\beta) = \sum_{i=1}^{m} \left[y_i \log(p(x_i)) + (Y - y_i) \log(Y - p(x_i)) \right]$$

$$p(x_i) = \frac{1}{1+e^{-z}} \quad (z = \beta_0 + \beta_1 x)$$

$$\frac{\partial L(\beta)}{\partial \beta_j} = \sum_{i=1}^{m} (\hat{y}_i - y_i) x^{(i)}$$

Same as linear regression.

Gradient ascent is same as gradient ascent just we have '+' sign.

Why should we take out derivative?

- The derivative of the log likelihood is taken with respect to the coefficients β_0 and β_1 to understand how changing the values of the coefficients affects the log likelihood.
- The optimization algorithm (e.g., gradient descent) uses the derivative of the log likelihood to determine the direction of the step that maximizes the log likelihood and iteratively updates the values of β_0 and β_1 until the log likelihood is maximized.
- The derivative of the log likelihood provides information about the rate of the change of the log likelihood with respect to the coefficients, which is used to find the values β_0 and β_1 that maximize the log likelihood and provide the best fit of the model of the data.
- From a geometrical perspective, the derivative of the log likelihood can be thought of as the slope of the log likelihood function. The optimization algorithm updates the values of β_0 and β_1 in the direction of the steepest ascent (ie, the direction of the highest slope) until the log likelihood is maximized.

Training with Gradient Descent

There's a way to use gradient DESCENT just like we used in LR but we need a bit of changes in our optimization problem, so we introductive NLL.
The negative log likelihood (NLL) is used to measure the goodness of fit of the logistic regression model to the data.
NLL is calculated by taking the negative logarithm of the likelihood function.
The goal in logistic regression is to find the values of beta that minimize the NLL, which is equivalent to maximizing the likelihood function.

negative log likelihood

Gradient Descent

$$NLL(\beta) = -\frac{1}{m}\left[\sum_{i=1}^{m} \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)\right]$$

Basically

$$NLL(\beta) = -L(\beta)$$

Want min $NLL(\beta)$:

repeat {

$$\beta_j = \beta_j - \alpha \sum_{i=1}^{m} (\hat{y}_i - y_i)\, x^{(i)}$$

Notice this is! Here we want to minimize.

Minimizing the negative log likelihood rather than maximizing the likelihood function has several advantages:

It's much easier to reason about the loss this way, to be consistent with the rule of loss functions approaching 0 as the model gets better.

Assumptions of Logistitic Regression

Binary response: The response variable should be binary, meaning it can take on only two values, such as 0 or 1.

Independence of observations: The observations should be independent of each other. This means that the outcomes of one observation should not affect the outcome of another observation.

Linearity in the log-odds: The log-odds of the response should have a linear relationship with the predictor variables. This means that the log-odds of the response can be estimated as a linear combination of the predictor variables.

Large sample size: The sample size should be large enough to accurately estimate the coefficients in the model.

For example, consider a study that examines the association between a person's smoking status (binary response) and their age (predictor variable). The independence of observations assumption would require that the outcome of one person's smoking status does not affect another person's smoking status. The linearity in the log-odds assumption would require that the log-odds of a person's smoking status can be estimated as a linear function of their age.