Module 2: Linear Model for Classification

By
Ms. Sarika Dharangaonkar
Assistant Professor,
KJSCE

Contents: Module2

Linear Basis Function Models

- Linear Basis Function Models, often referred to simply as linear models, are a class of machine learning algorithms used for various tasks like regression and classification.
- These models involve transforming the input data using a set of basis functions to create a linear combination of features, which then forms the foundation for making predictions or classifications.

Key Components of Linear Basis Function Model

- **Basis Functions:**
  - Basis functions are functions that transform the input features into a new representation.
  - Common examples include polynomial basis functions (e.g., quadratic, cubic), Gaussian radial basis functions, and Fourier basis functions.
  - The choice of basis functions impacts how well the model can capture the underlying patterns in the data.
- **Weights and Parameters:**
  - Linear models include weights or coefficients associated with each basis function.
  - These weights determine the contribution of each basis function to the final prediction.
  - Adjusting these weights during the learning process is essential for the model to fit the data effectively.

Key Components of Linear Basis Function Model

- **Linear Combination:**
  - Linear models combine the basis functions and their associated weights in a linear manner to produce the model's output.
  - The transformed features are linearly weighted to generate predictions.

Types of Linear Basis Function Models

1. Linear Regression:

- Linear regression is a classic example of a linear basis function model.
- It uses a linear combination of input features (possibly transformed by basis functions) to predict a continuous target variable.

2. Logistic Regression:

- Despite the name, logistic regression is a classification algorithm that uses linear combinations of features (often transformed by sigmoid functions) to estimate the probability of a binary outcome.

3. Ridge Regression (L2 Regularization):

- Ridge regression extends linear regression by adding a regularization term based on the L2 norm of the weights.
- This helps prevent overfitting and can handle multicollinearity between features.

Types of Linear Basis Function Models…

4. Lasso Regression (L1 Regularization):

- Lasso regression is similar to ridge regression, but it adds an L1 regularization term to the cost function.
- Lasso can also perform feature selection by driving some weights to exactly zero.

5. Polynomial Regression:

- Polynomial regression is a form of linear regression where the input features are transformed using polynomial basis functions, allowing the model to capture non-linear relationships.

Advantages of linear basis function models:

- Linear models are computationally efficient and can be trained on large datasets.
- They provide interpretable results, as the impact of each feature can be directly observed through the weights.
- Linear models can generalize well when the data exhibits linear or near-linear relationships.

Limitations of linear basis function models:

- Linear models struggle with capturing complex non-linear relationships in the data.
- The effectiveness of linear models highly depends on the choice of appropriate basis functions.
- For high-dimensional datasets, linear models might not perform well due to their simplicity.

Linear Regression

- The process of finding a straight line that best approximates a set of points on graph is called Linear Regression
- The word Linear signifies that the type of relationship that you try to establish between the variables tends to be a straight line
- It is simple and popular technique of regression analysis

Example:

- When you visit the a petrol pump, you have good estimate of how much you would drive in the next few days and how much petrol you will need correspondingly?
- Here how you are predicting your usage? It is based on your past data points. That is nothing but the regression analysis

Example…

- When we try to establish a correlation between one variable (independent variable) with another variable (dependent variable) and use the relationship to predict the values.
- So, we can write:
  - Petrol required (X)= 5litres
  - Kilometres driven (Y)= 50km
  - Then we can write the relationship as: Y= 10X
- Prediction: For 10litres of Petrol you can go up to 10x10=100kms

Types of Linear Regression

- Simple Linear Regression (SLR):
  - It has only one Independent Variable.
  - Example: No. of litres of petrol required and Kilometres driven.
- Multiple Linear Regression (MLR):
  - It has more than one Independent(or Input) Variables.
  - Example: Number of litres of petrol, age of vehicle, speed and kilometres driven.

- General Formula for Linear Regression is:

  $Y = Bo + B1X1 + B2X2 + B3X3 + B4X4 \ldots\ldots + BiXi + E$

Where,

- Y is the outcome(dependent variable)
- Xi are the values of independent variables
- Bo is the value of Y when each Xi=0. It is also called as Y intercept.
- Bi is the change in Y based on the unit change in Xi. It is called as regression coefficient or slope of the regression line.
- E is the random error or noise that represents the difference between the predicted value and actual values.

Linear Regression Algorithm Steps:

- **Data Collection:**
  - Collect a dataset that includes the pairs of independent variable(s) (x) and dependent variable (y) you want to model.
- **Data Preprocessing:**
  - Clean the data, handle missing values, and remove outliers if necessary. Preprocess the data to ensure it's ready for analysis.
- **Data Splitting:**
  - Divide the dataset into training and testing sets. A common split is around 70-80% of the data for training and the rest for testing.
- **Model Initialization:**
  - Initialize the model by assuming an initial value for the slope (m) and the intercept (c) of the linear equation: $y = mx + c$.

- **Cost Function:**
  - Define a cost function that quantifies the difference between the predicted values and the actual values in the training set.
  - A commonly used cost function for linear regression is Mean Squared Error (MSE).
- **Gradient Descent:**
  - Use an optimization algorithm like gradient descent to minimize the cost function.
  - This involves iteratively updating the values of m and c to find the values that minimize the cost function and best fit the data.
- **Model Training:**
  - Iterate through the training dataset, calculate the predicted values using the current values of m and c, compute the gradients of the cost function with respect to m and c, and update the values of m and c using the gradients and a learning rate.

- **Making Predictions:**
  - After training, use the learned values of m and c to make predictions on new or unseen data.
  - Plug the independent variable(s) into the linear equation to get predicted values of the dependent variable.
- **Model Evaluation:**
  - Evaluate the performance of the trained model using the testing dataset.
  - Common evaluation metrics include Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared.
- **Visualization:**
  - Visualize the regression line along with the data points to understand how well the model fits the data.

Numerical on Linear Regression

- Suppose you are given a dataset that represents the relationship between the number of hours studied and the corresponding exam scores. You want to build a linear regression model to predict exam scores based on the number of hours studied.
- Here's a small portion of the dataset:

Hours Studied (x)	Exam Score (y)
2	50
3	65
4	75
5	80
6	90

Solution:

- To find the linear relationship between hours studied and exam scores the formula can be given as: $y = ax + b$

Where:

y is the predicted exam score

x is the number of hours studied

m is the slope (coefficient) of the line

b is the y-intercept

Step 1: Calculate the means of x and y:

mean(x) = (2 + 3 + 4 + 5 + 6) / 5 = 4

mean(y) = (50 + 65 + 75 + 80 + 90) / 5 = 72

Step 2: Calculate the slope (a):

a = Σ((x - mean(x)) * (y - mean(y))) / Σ((x - mean(x))^2)

= ((2-4)*(50-72) + (3-4)*(65-72) + (4-4)*(75-72) + (5-4)*(80-72) + (6-4)*(90-72)) / ((2-4)^2 + (3-4)^2 + (4-4)^2 + (5-4)^2 + (6-4)^2)

= 485 / 10

= 48.5

Step 3: Calculate the y-intercept (b):

b = mean(y) - m * mean(x) = 72 - 48.5 * 4 = -18

So, the equation for the linear regression line is:

y = 48.5x - 18

- Now you can use this equation to predict exam scores for different values of hours studied.
- For example, if someone studies for 7 hours:
- y = 48.5 * 7 - 18 = 337.5

Example 2:

Problem:

Suppose you are given a dataset that contains information about the size of houses (in square feet) and their corresponding prices (in thousands of dollars). You want to build linear regression model to predict house prices based on their size.

Here's a portion of the dataset:

House Size (x)	Price (y)
1400	250
1600	300
1700	320
1870	370
2200	450

Example 3:

Problem:

Consider a dataset that contains information about the number of years of work experience and the corresponding salary in thousands of dollars. You want to build a linear regression model to predict salary based on the number of years of work experience.

Here's a portion of the dataset:

Years of Experience (x)	Salary (y)
1	40
2	50
3	60
4	65
5	75

Logistic Regression

- Logistic Regression is classification algorithm
- It's used to predict the probability that an input belongs to a particular class.
- Sometime, the output(the dependent variable) of the regression is binary event having yes/no, pass/fail, 0/1 kind of results.
- In this case you need a regression model that find the probability of the event occurring or not.
- Example:
  - Given a particular humidity of the day, what are the chances of raining?
  - Given certain credit score, what are the possibilities of getting your loan application approved?

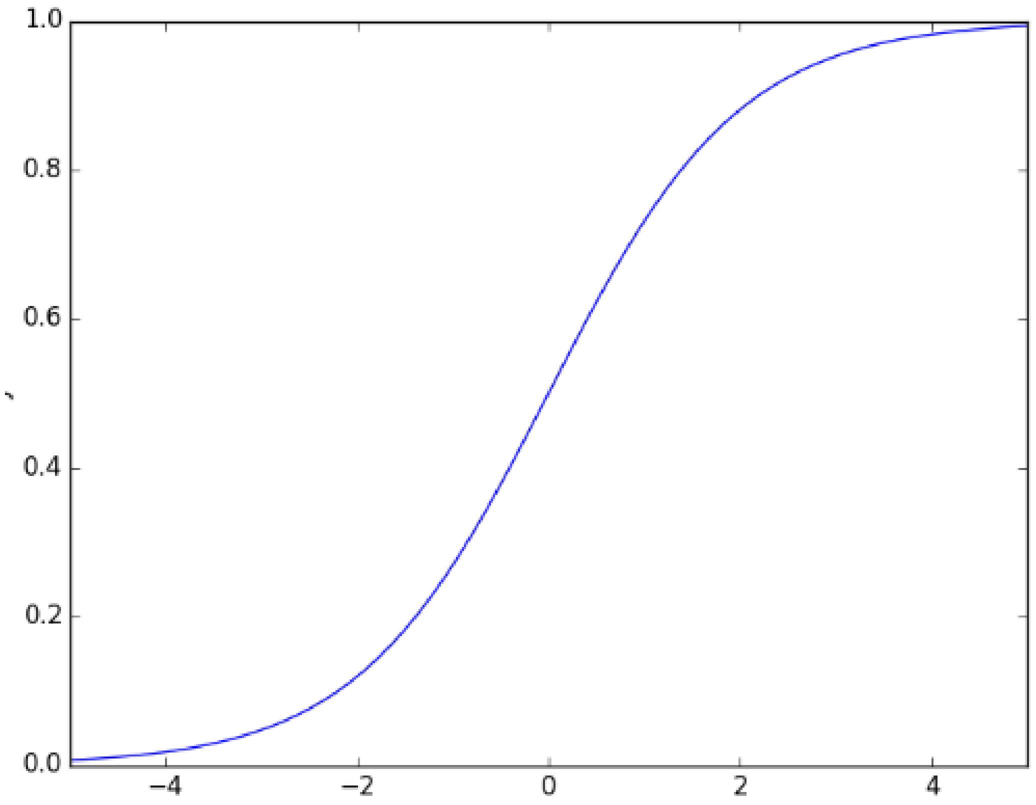
Logistic Regression provides a probability prediction model based on the values of independent variables.

Logistic Regression

- Logistic regression is a regression technique used to model and estimate the probability of an event to occur based on the values of the independent variables.
- Unlike Linear regression line, the logistic regression line is a curve.
- This curve is called as Sigmoid curve or S-curve
- All the outcome data points are either concentrated on 0 or 1.

Sigmoid Curve of Logistic Regression

- The curve depicts various possible probabilities of an event, occurring between 0(totally uncertain) and 1(totally certain), on the y-axis.



Logistic Regression..

- The Probability prediction Formula for logistic regression is as following:

$$\hat{P} = \frac{e^{\beta_0 + \beta_1 x_1}}{1 + e^{\beta_0 + \beta_1 x_1}}$$

- Where, p is predicted probability
- e is the exponential function
- Bo and B1 are regression coefficients
- X1 is the value of the independent variable

Numerical on Logistic Regression…

- A team scored 285 runs in a cricket match. Assuming regression coefficients to be 0.3548 and 0.00089 respectively, calculate its probability of winning the match.

The logistic regression coefficients are given to be $\beta_0 = 0.3548$ and $\beta_1 = 0.00089$ and the value of the independent variable (match score) $X_1 = 285$. Putting these values in the logistic regression formula.

$$\hat{p} = \frac{e^{\beta_0 + \beta_1 x_1}}{1 + e^{\beta_0 + \beta_1 x_1}}$$

$$\hat{p} = \frac{e^{0.3548 + 0.00089 \times 285}}{1 + e^{0.3548 + 0.00089 \times 285}} = \frac{1.8375}{1 + 1.8375}$$

$$= 0.6475$$

Hence, the probability of the team winning the match is 0.6475 or 64.75%.

Bayesian Linear Regression

- Bayesian Linear Regression is an extension of the traditional linear regression that incorporates Bayesian principles to estimate the parameters of the linear regression model.
- It combines prior knowledge (prior distribution) about the parameters with observed data to obtain a posterior distribution, allowing for uncertainty quantification and better inference about the model's parameters.
- Bayesian Linear Regression provides a probabilistic framework for modeling uncertainty in parameter estimates, making it particularly useful when dealing with limited data or when you want to quantify uncertainty in your predictions.
- It requires setting up appropriate prior distributions and often involves more computation compared to traditional linear regression.

Bayesian Linear Regression…

Model Assumption:

- Similar to standard linear regression, the basic assumption is that the relationship between the input features (predictors) and the target variable (response) is linear.

Parameter Estimation in Standard Linear Regression:

- In traditional linear regression, parameter estimation involves finding the values of the coefficients that minimize the sum of squared differences between predicted and actual values (ordinary least squares).
- This estimation does not provide information about the uncertainty associated with the parameter estimates.

Bayesian Linear Regression…

Bayesian Approach:

- Bayesian Linear Regression treats the coefficients as random variables with probability distributions (prior distributions).
- Prior distribution reflects the initial belief or knowledge about the parameter values before observing any data.

Bayes' Theorem:

- Bayesian inference involves using Bayes' theorem to update our beliefs about the parameters after observing the data.
- **Posterior distribution = (Likelihood * Prior) / Evidence**, where:
  - Likelihood: Measures how well the observed data matches the predictions made by the model.
  - Prior: Reflects prior knowledge or beliefs about the parameter values.
  - Evidence: Acts as a normalization factor, ensuring that the posterior distribution is a proper probability distribution.

Bayesian Linear Regression…

Parameter Estimation with Bayesian Linear Regression:

- The posterior distribution gives us a range of possible parameter values along with their probabilities.
- The mean of the posterior distribution represents the point estimate of the parameters, and the spread of the distribution represents the uncertainty associated with the estimates.

Prediction and Uncertainty:

- Bayesian Linear Regression provides not only point predictions but also predictive distributions for new data points.
- Prediction intervals or credible intervals from the posterior distribution can be used to quantify prediction uncertainty.

Bayesian Linear Regression…

- Bayesian Linear Regression provides a probabilistic framework for modeling uncertainty in parameter estimates, making it particularly useful when dealing with limited data or when you want to quantify uncertainty in your predictions.
- It requires setting up appropriate prior distributions and often involves more computation compared to traditional linear regression.

Bayesian Linear Regression step by step

Step 1: Model Assumption

- Bayesian Linear Regression assumes that the relationship between the input features ‘X’ and the target variable ‘y’ is linear. The standard linear regression equation is given by:

$$y = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \ldots + \beta_n * x_n + \varepsilon$$

- Where:
- y is the target variable (response).
- $x_1, x_2, \ldots, x_n$ are the input features.
- $\beta_0, \beta_1, \ldots, \beta_n$ are the coefficients (parameters) we want to estimate.
- ε represents the error term.

Step 2: Introduction of Bayesian Principles

- In Bayesian Linear Regression, we treat the coefficients as random variables with prior distributions that reflect our initial beliefs about their values.

Step 3: Prior Distribution

- We introduce prior distributions for the coefficients to represent our prior beliefs about their values. For simplicity, let's assume Gaussian priors for each coefficient:

$$\beta_i \sim N(\mu_0, \sigma_0^2)$$

- Where:
- β_i is the i-th coefficient.
- $N(\mu_0, \sigma_0^2)$ represents a Gaussian (normal) distribution with mean μ_0 and variance σ_0^2.

Step 4: Likelihood Function

- The likelihood function quantifies how well the observed data fits the model's predictions. Assuming the errors (ε) are normally distributed with mean 0 and variance σ^2, the likelihood can be written as:

$$p(y \mid X, \beta) = N(X\beta, \sigma^2 I)$$

Where:

- y is the observed target variable.
- X is the matrix of input features.
- β is the vector of coefficients.
- σ^2 is the error variance.
- I is the identity matrix.

Step 5: Posterior Distribution

- Using Bayes' theorem, the posterior distribution is proportional to the product of the prior and likelihood:

$$p(\beta \mid X, y) \propto p(y \mid X, \beta) * p(\beta)$$

Step 6: Compute the Posterior

- For simplicity, let's assume a conjugate prior (a prior that leads to a posterior in the same distribution family) and choose a Gaussian prior for β. The posterior distribution for each coefficient is also Gaussian:

$$\beta_i \mid X, y \sim N(\mu_i, \sigma_i^2)$$

Where:

- $\mu_i = (\sigma^2{}_0 * X^T X + \sigma^2 * I)^{-1} * (\sigma^2{}_0 * \mu_0 + X^T y)$
- $\sigma_i^2 = (\sigma^2{}_0 * X^T X + \sigma^2 * I)^{-1}$

Step 7: Make Predictions

- To make predictions for new data, use the estimated posterior distributions of the coefficients. The predictive distribution for a new observation x_new is:

$$p(y\_new \mid x\_new, X, y) = N(\mu\_new, \sigma^2\_new + \sigma^2)$$

Where:

- $\mu\_new = x\_new^T \mu$
- $\sigma^2\_new = x\_new^T \sigma^2 X(x\_new^T X + \sigma^2 I)^{-1} x\_new$

Step 8: Hyperparameters and Inference

Hyperparameters like μ_0, σ_0^2, and σ^2 control the shape of the prior distributions and the amount of regularization. These can be chosen based on domain knowledge or cross-validation.

Discriminant Function

- A discriminant is a function that takes an input vector **x** and assigns it to one of *K* classes, denoted *Ck*.
- The simplest representation of a linear discriminant function is obtained by taking a linear function of the input vector so that

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$$

- where **w** is called a *weight vector*, and *w*0 is a *bias*
- The negative of the bias is sometimes called a *threshold*.
- An input vector **x** is assigned to class *C*1 if *y*(**x**) 0 and to class *C*2 otherwise.

Discriminant Function…

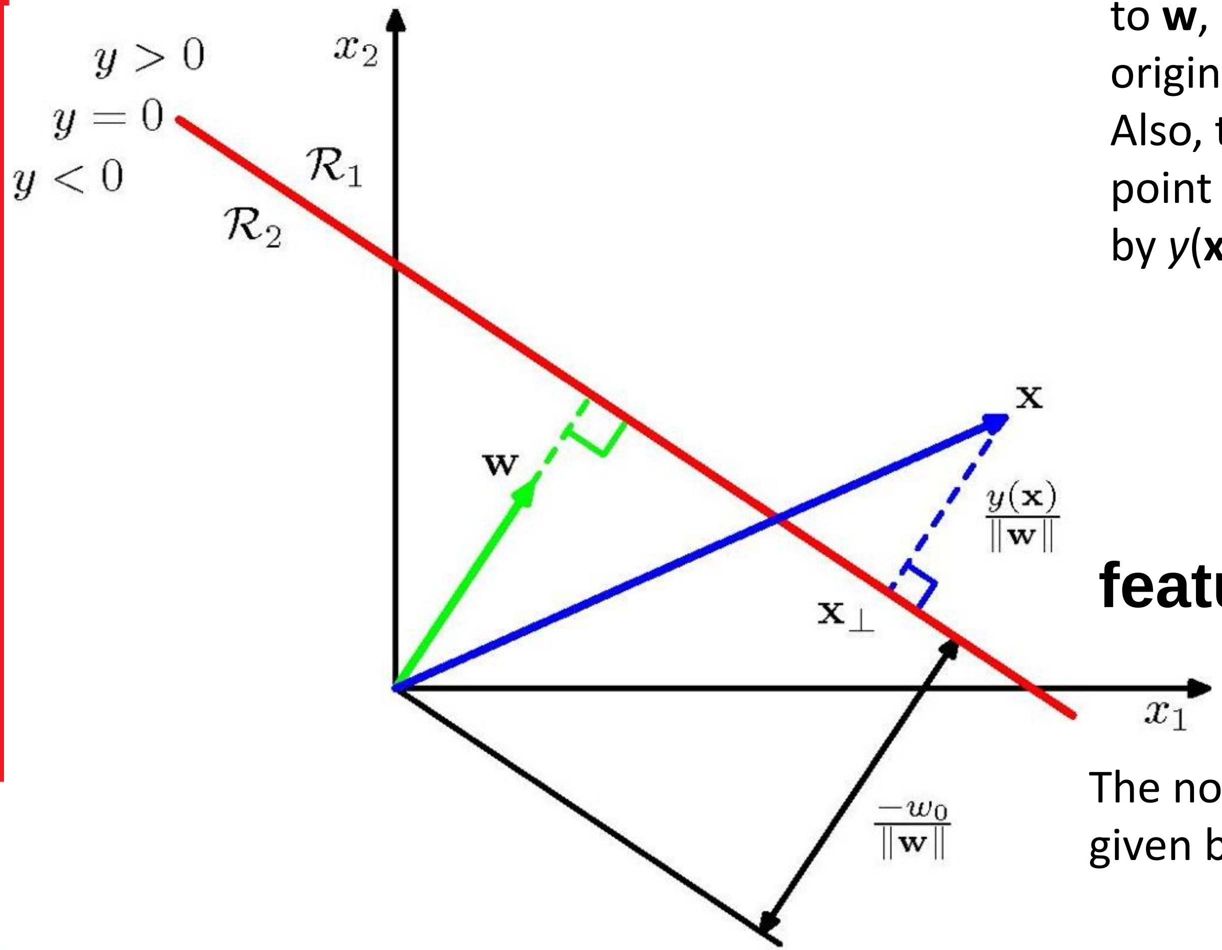


Illustration of the geometry of a linear discriminant function in two dimensions. The decision surface, shown in red, is perpendicular to **w**, and its displacement from the origin is controlled by the bias parameter *w*0. Also, the signed orthogonal distance of a general point **x** from the decision surface is given by *y*(**x**)/**w**.

- w determines orientation of decision boundary
- ω_0 determines location of decision boundary

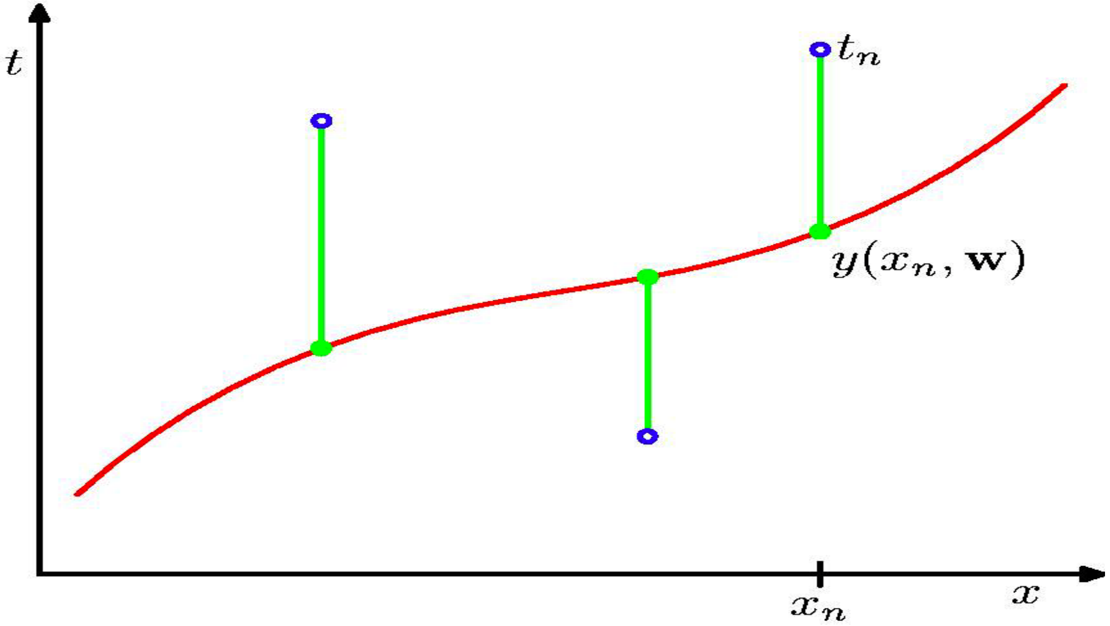
The normal distance from the origin to the decision surface is given by

$$\frac{\mathbf{w}^{\mathrm{T}}\mathbf{x}}{\|\mathbf{w}\|} = -\frac{w_0}{\|\mathbf{w}\|}.$$

Discriminant functions - How to determine parameters?

1. Least Squares for Classification

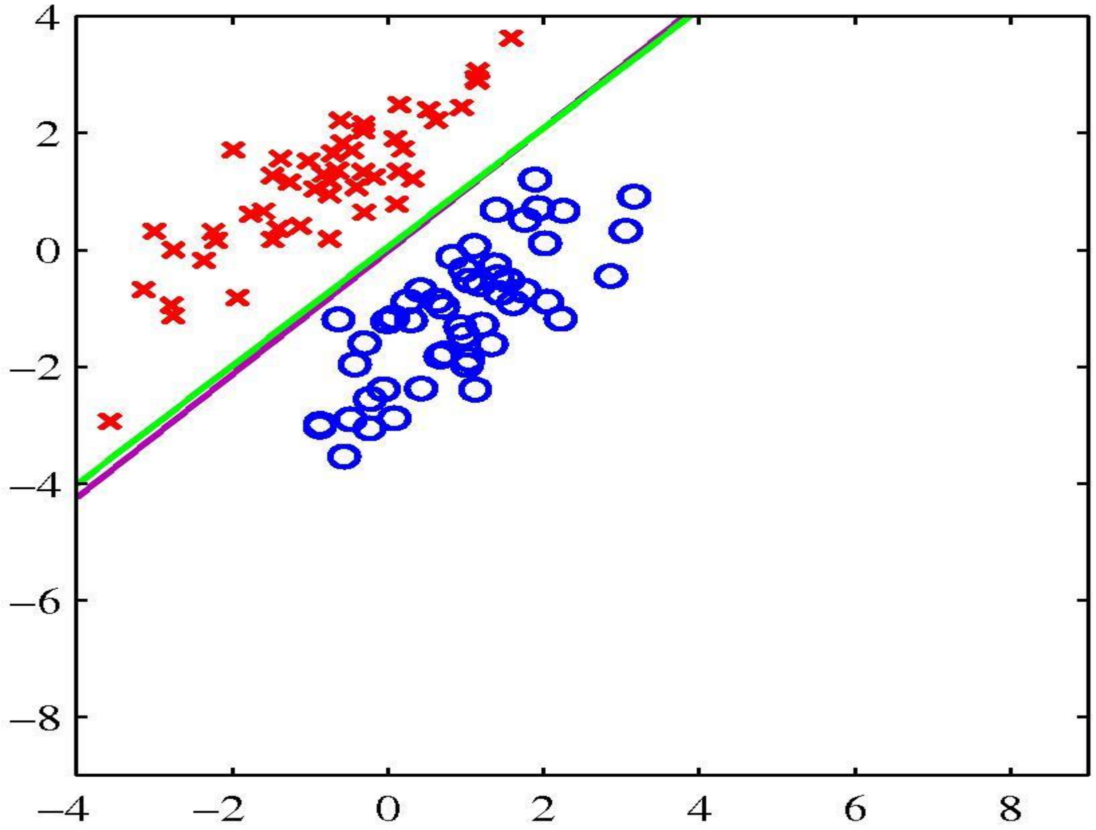
- <u>General Principle:</u> Minimize the squared distance (residual) between the observed data point and its prediction by a model function



$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2$$

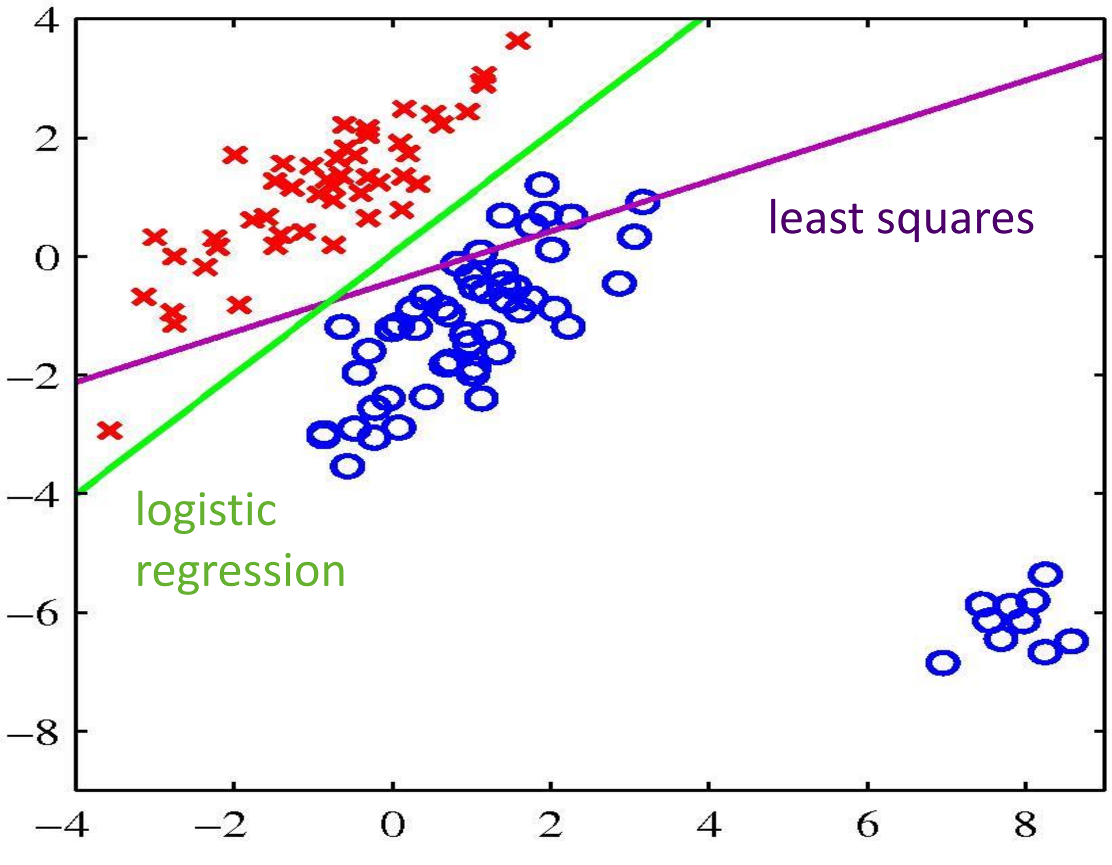
Discriminant functions - How to determine parameters?

- In the context of classification: find the parameters which minimize the squared distance (residual) between the data points and the decision boundary



Discriminant functions - How to determine parameters?

- Problem: sensitive to outliers; also distance between the outliers and the discriminant function is minimized --> can shift function in a way that leads to misclassifications

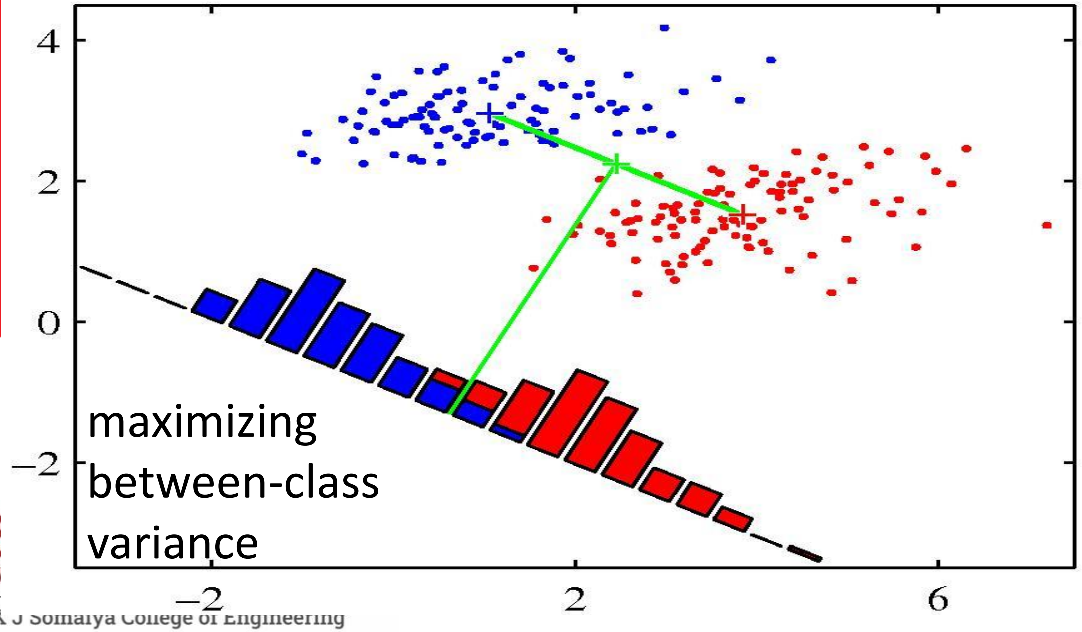


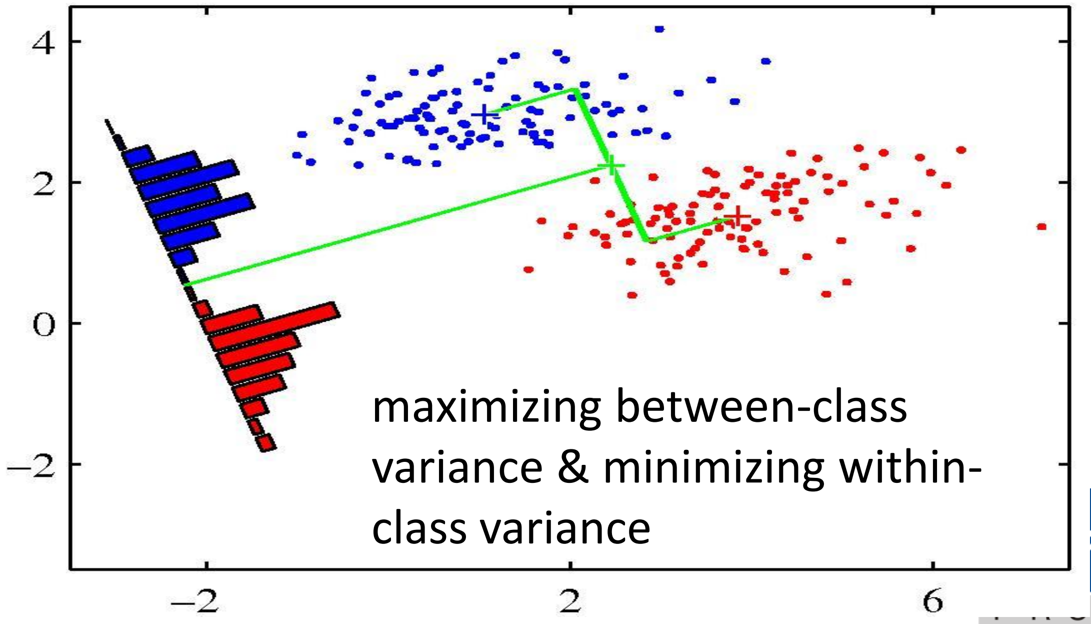
Discriminant functions - How to determine parameters?

2. Fisher's Linear Discriminant

- General Principle: Maximize distance between means of different classes while minimizing the variance within each class

$$J(w) = \frac{|m_1 - m_2|^2}{s_1^2 + s_2^2}$$





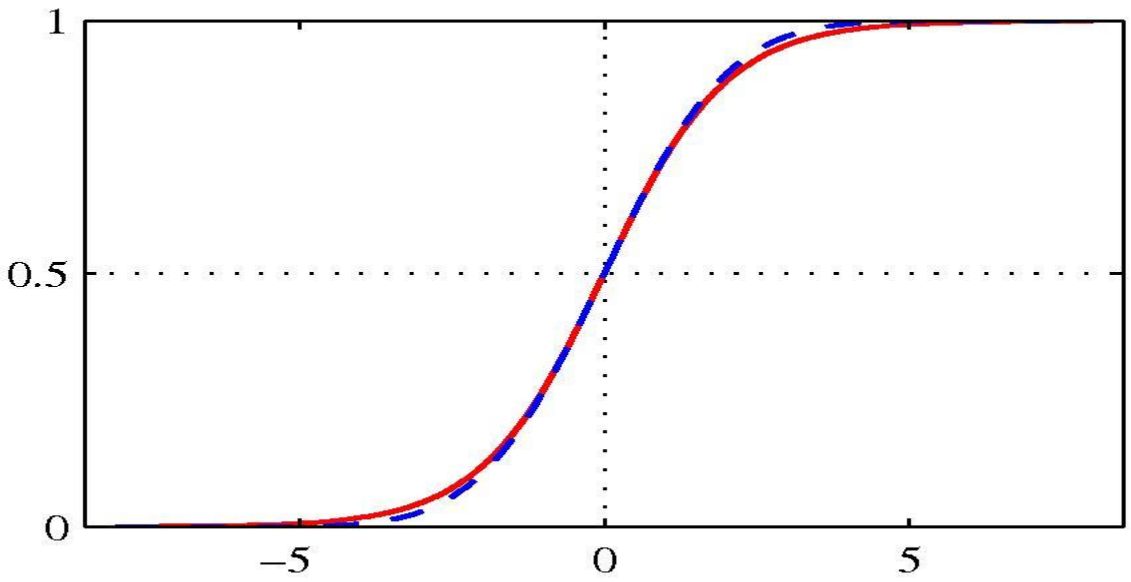
Discriminant Function

- A discriminant function focuses on finding a decision boundary that separates different classes in the feature space.
- In a two-dimensional feature space, you can visualize this as a line or curve that separates the data points belonging to different classes.
- For example, in a binary classification scenario, you might have two classes, and the discriminant function helps determine whether a new data point should be classified as one class or the other based on which side of the decision boundary it falls.

```
Class A
    |
---|---- Decision Boundary
    |
Class B
```

Probabilistic Generative Models

- Model class-conditional densities ($p(x \mid C_k)$) and class priors ($p(C_k)$)
- Use them to compute posterior class probabilities ($p(C_k \mid x)$) according to Bayes theorem
- Posterior probabilities can be described as logistic sigmoid function



Inverse of sigmoid function is the logit function which represents the ratio of the posterior probabilities for the two classes

$\ln[p(C_1 \mid x)/p(C_2 \mid x)]$ --> log odds

Probabilistic Generative Model

Gaussian Mixture Models (GMMs):

- Gaussian Mixture Models assume that the data comes from a mixture of several Gaussian distributions. The model involves estimating the parameters of these Gaussians.
- Equation for Gaussian distribution:
  - $p(x|\mu, \sigma^2) = (1 / \sqrt{(2\pi\sigma^2)}) * \exp(-(x - \mu)^2 / (2\sigma^2))$
  - In the case of GMM, the data point x is generated from one of the K Gaussian components with probability π_k:
  - $p(x) = \Sigma[\pi\_k * p(x|\mu\_k, \sigma^2\_k)]$ for $k = 1$ to K
- Parameters to be learned:
- μ_k: Mean of the k-th Gaussian component.
- σ^2_k: Variance of the k-th Gaussian component.
- π_k: Mixing coefficient representing the probability of choosing the k-th component.

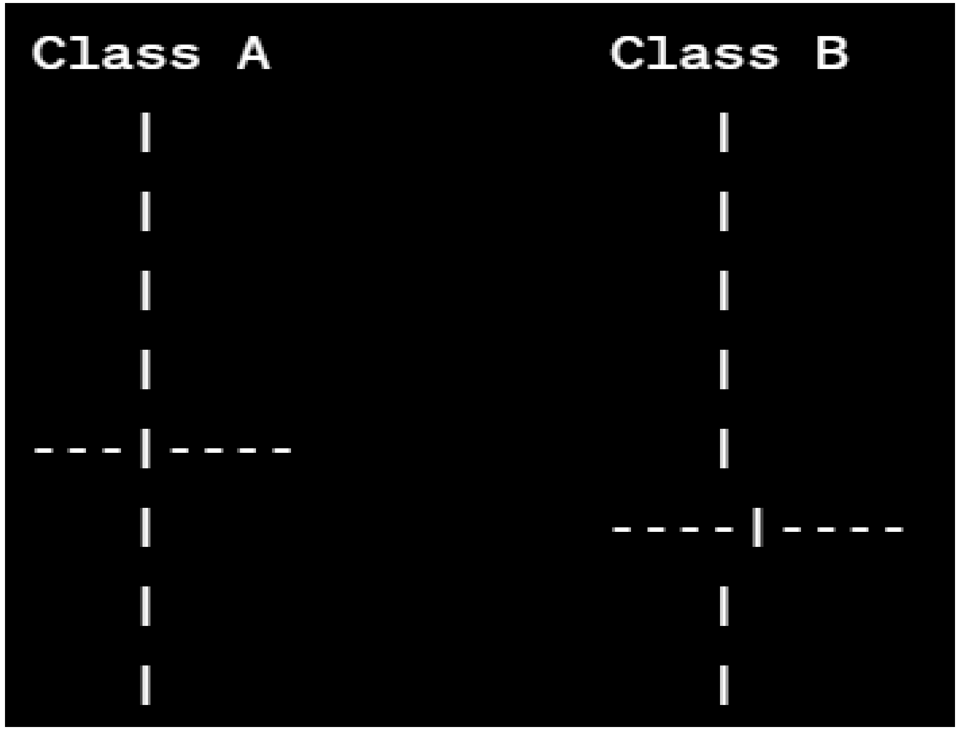
Probabilistic Generative Model

Naive Bayes Classifier:

- Naive Bayes Classifier is a generative model that assumes independence among features given the label. The model calculates the posterior probability of the label given the features using Bayes' theorem:
- $P(y|x_1, x_2, ..., x\_n) \propto P(y) * P(x_1|y) * P(x_2|y) * ... * P(x\_n|y)$
- Parameters to be learned:
- P(y): Prior probability of each label.
- $P(x_i|y)$: Probability of observing feature x_i given label y.

Probabilistic Generative Model:

- A probabilistic generative model aims to model the distribution of data for each class separately and then uses these distributions to generate new data points.
- In a two-class scenario, you could visualize this as two overlapping distributions (e.g., Gaussian distributions) for each class, indicating the probability density for different feature values.
- The overlap of these distributions determines the region where the model might be uncertain about the class.



Probabilistic Discriminative Models

- Probabilistic Discriminative Models are a class of machine learning models that focus on directly modeling the conditional probability of the target variable (output) given the input features.
- These models are designed to classify data points into different classes based on the learned relationship between the features and the target variable.
- Unlike generative models, which aim to learn the full data distribution, discriminative models are primarily concerned with making accurate predictions and decisions.

Probabilistic Discriminative Models

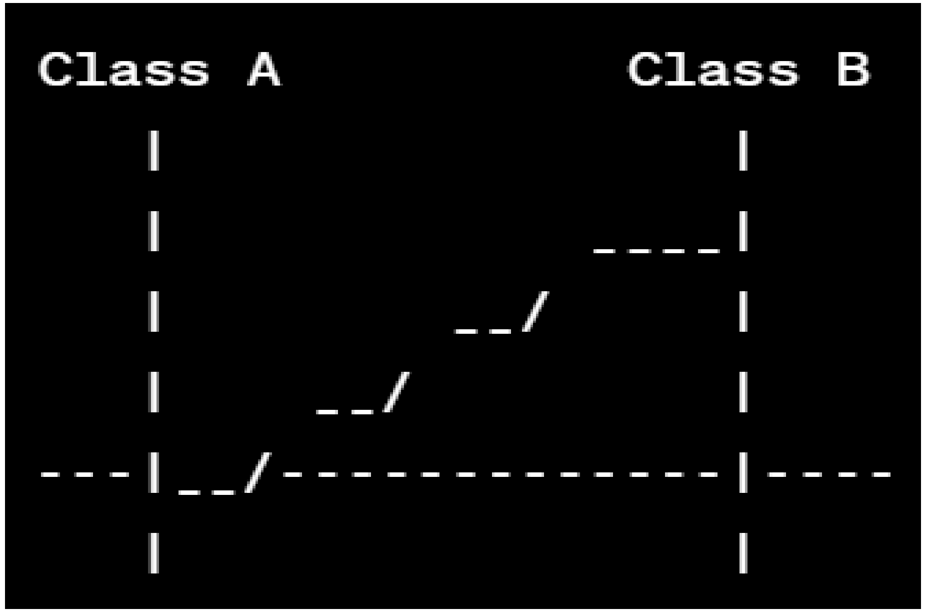
Types of Probabilistic Discriminative Models:

Several types of probabilistic discriminative models exist:

- **Logistic Regression:**
  - Models the probability of a binary outcome using the logistic function. It's a fundamental discriminative model for binary classification.
- **Linear Discriminant Analysis (LDA):**
  - Assumes that the data follows a Gaussian distribution and models the decision boundary using a linear combination of input features.
- **Support Vector Machines (SVMs):**
  - Maximizes the margin between different classes by finding the hyperplane that best separates the data.
- **Neural Networks:**
  - Can be designed as discriminative models by using appropriate activation functions in the output layer and optimizing them using techniques like backpropagation.

Probabilistic Discriminative Model:

- A probabilistic discriminative model focuses on modeling the probability of each class given the input features.
- In a two-dimensional feature space, this could be represented as contours that indicate the probability of a data point belonging to a certain class.
- The decision boundary is where these contours intersect, indicating a threshold at which the model makes a decision about the class.



Aspect	Probabilistic Generative Models	Probabilistic Discriminative Models
Main Focus	Model joint distribution of features and classes	Model conditional distribution of classes given features
Goal	Provide a complete probabilistic description of data	Learn decision boundaries for classification tasks
Usage	Generate new samples, impute missing data, anomaly detection	Classification tasks with strong performance
Performance vs. Classification	May not achieve the same classification performance as discriminative models	Often provide better classification performance
Examples	Gaussian Mixture Models (GMMs), Naive Bayes, Hidden Markov Models (HMMs)	Logistic Regression, Support Vector Machines (SVMs), Conditional Random Fields (CRFs), Neural Networks
Data Requirement	Require a relatively larger amount of data	Can perform well even with smaller datasets
Computational Complexity	Generally more complex due to modeling joint distribution	Often computationally more efficient
Data Generation vs. Classification	Can generate new samples similar to training data	Focused on classifying new data points
Task Emphasis	Emphasis on understanding data generation process	Emphasis on accurate classification

Aspect	Discriminant Function	Probabilistic Generative Models	Probabilistic Discriminative Models
Main Purpose	Compute decision boundary directly	Model joint distribution of data	Model conditional distribution of classes
Output	Scalar value indicating the class	Probabilities for classes and data	Probabilities for classes given data
Example	Linear discriminant function (LDA), Support Vector Machines (SVMs)	Gaussian Mixture Models (GMMs), Naive Bayes	Logistic Regression, Neural Networks
Training	Learns decision boundary directly from data	Models data generation process	Models decision boundary for classification
Data Generation	Does not generate new data	Can generate new samples	Does not generate new data
Probability Estimation	Does not provide explicit probability estimates	Provides explicit class and data probabilities	Provides class probabilities given data
Use Case	Binary classification, multi-class classification	Image generation, data augmentation	Image classification, sentiment analysis
Performance	Good for simple data distributions, can work well with limited data	May not be optimal for complex distributions, but provides insights into data generation process	Often achieves strong classification performance
Example	Given a new data point, computes a score that determines the class	Given data and classes, models how the data is generated	Given data, models how classes are distributed
Examples	Linear Discriminant Analysis (LDA), Support Vector Machines (SVMs)	Gaussian Mixture Models (GMMs), Naive Bayes	Logistic Regression, Neural Networks

Sr.No	Linear Regression	Logistic Regression
1	Linear regression is used to predict the continuous dependent variable using a given set of independent variables.	Logistic regression is used to predict the categorical dependent variable using a given set of independent variables.
2	Linear regression is used for solving Regression problem.	It is used for solving classification problems.
3	In this we predict the value of continuous variables	In this we predict values of categorical varibles
4	In this we find best fit line.(Linear)	In this we find Sigmoid-Curve .
5	Least square estimation method is used for estimation of accuracy.	Maximum likelihood estimation method is used for Estimation of accuracy.
6	The output must be continuous value,such as price,age,etc.	Output is must be categorical value such as 0 or 1, Yes or no, etc.
7	It required linear relationship between dependent and independent variables.	It not required linear relationship.
8	There may be collinearity between the independent variables.	There should not be collinearity between independent variable.

Thank You