# Modelling

CSC4601 Theory of Machine Learning

# Today’s Plan

1. What is a model?

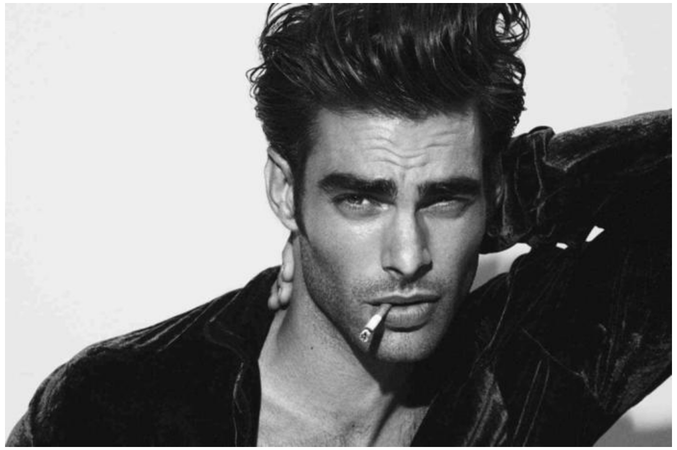
2. What is a linear model?

3. What is a linear model for regression and classification?

4. Advantages and Disadvantages of a linear model

# What is a model?

# Models



# Models

# (Mathematical) Models

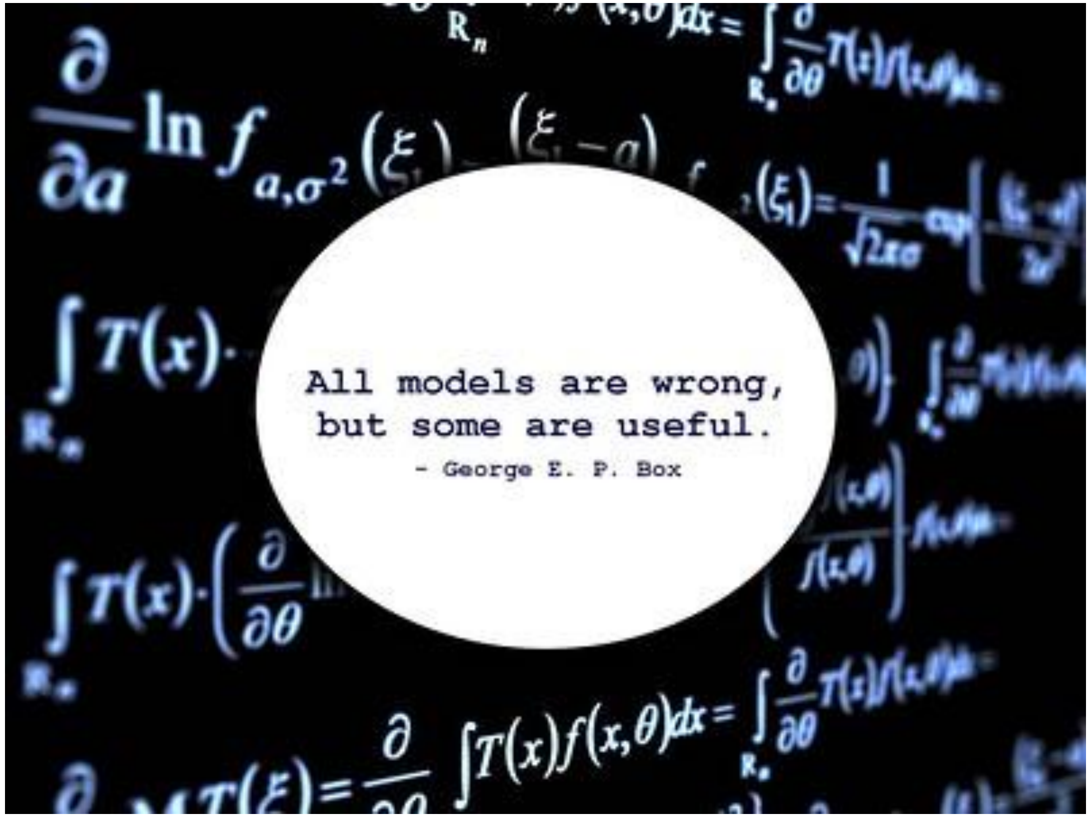
- A tool used to mathematically represent some observed phenomena.

- Uses relationships (math) to:
  - Explain a system
  - Encode system behavior
  - Make predictions about behavior

- A model consists of finding a relationship or mapping between an input variable and output variable.

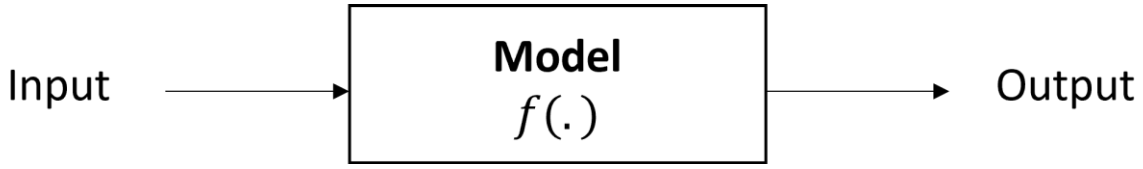
# Models

- Models are abstractions and simplification of the real world:
  - Modeling is a mathematical way to represent how we think about observed phenomena!



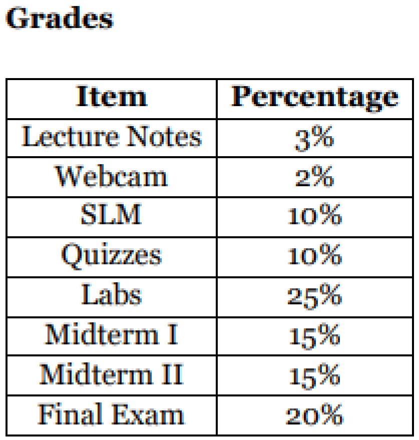
# Models : Input-Output Relationship

- One way to mathematically represent a model is through an input-output relationship.
- A model can be described as a mathematical formula that takes in an input variable and transforms it into an output variable.



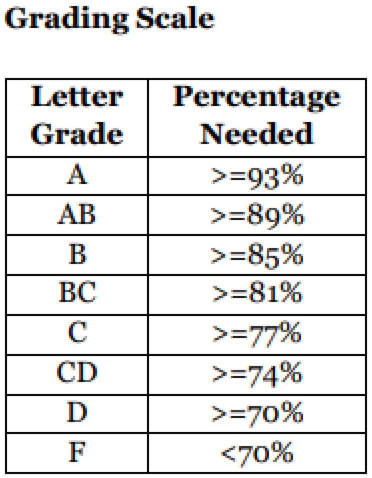
# Model Example 1

- Grades in a class
  - Input:
  - Model Parameters:
  - Model output:



**Grades**

| Item | Percentage |
|---|---|
| Lecture Notes | 3% |
| Webcam | 2% |
| SLM | 10% |
| Quizzes | 10% |
| Labs | 25% |
| Midterm I | 15% |
| Midterm II | 15% |
| Final Exam | 20% |

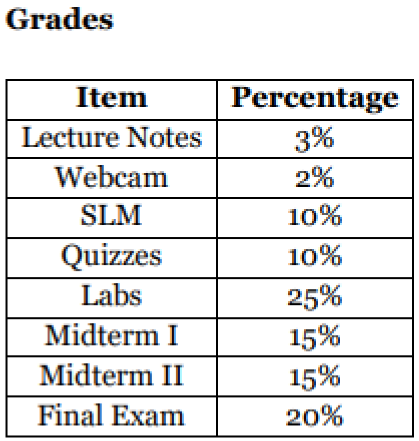


**Grading Scale**

| Letter Grade | Percentage Needed |
|---|---|
| A | >=93% |
| AB | >=89% |
| B | >=85% |
| BC | >=81% |
| C | >=77% |
| CD | >=74% |
| D | >=70% |
| F | <70% |

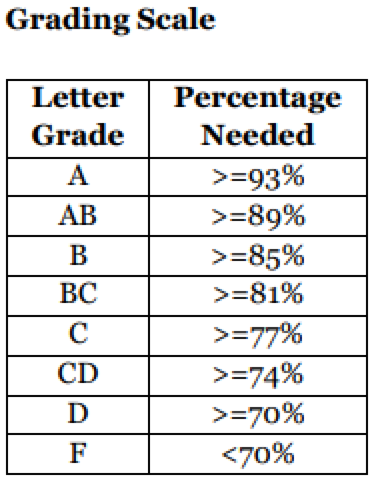
# Model Example 1

- Grades in a class
  - Input:
    - Score earned on assessments
  - Model Parameters:
    - Weighting of grades by type.
      - Quizzes – 10%
      - Midterm I – 15%
      - Etc.
  - Model output:
    - Grade Letter: A, AB, B, BC…



**Grades**

| Item | Percentage |
|---|---|
| Lecture Notes | 3% |
| Webcam | 2% |
| SLM | 10% |
| Quizzes | 10% |
| Labs | 25% |
| Midterm I | 15% |
| Midterm II | 15% |
| Final Exam | 20% |



**Grading Scale**

| Letter Grade | Percentage Needed |
|---|---|
| A | >=93% |
| AB | >=89% |
| B | >=85% |
| BC | >=81% |
| C | >=77% |
| CD | >=74% |
| D | >=70% |
| F | <70% |

# Model Example 2

- Aligning Satellite Images
  - Input:
  - Model Parameters:
  - Model output:

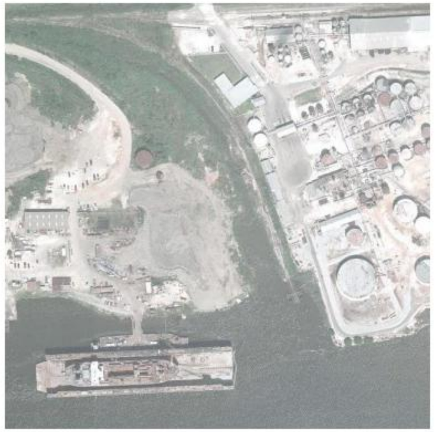


Image $A$

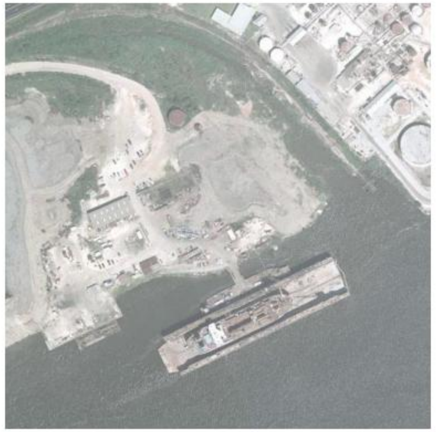
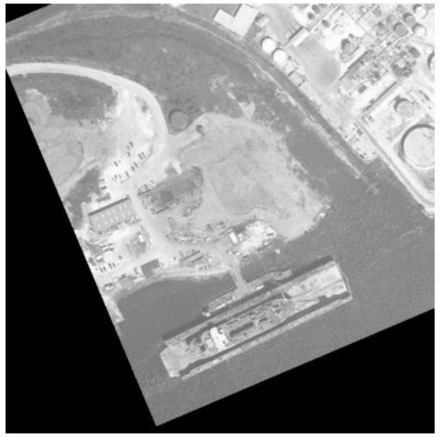


Image $B$

$$u = f_1(x, y)$$
$$v = f_2(x, y)$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



Mapped $A$

Original $B$

# Model Example 2

- Aligning Satellite Images
  - Input:
    - Image
  - Model Parameters:
    - *a,e: Scaling terms*
    - *a,b,d,e: Rotation terms*
    - *c, f: Translation terms*
  - Model output:
    - Transformed Image

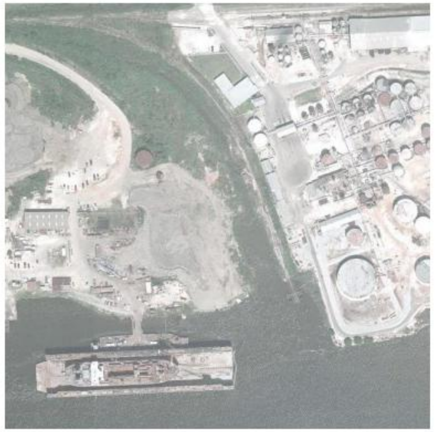


Image $A$

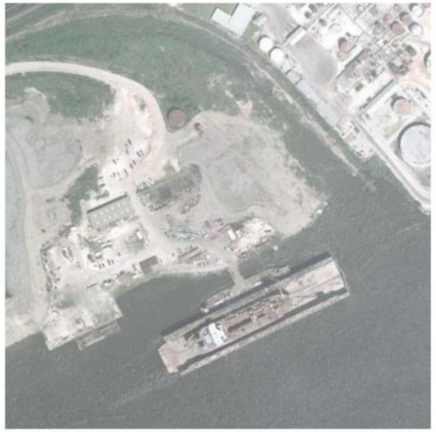
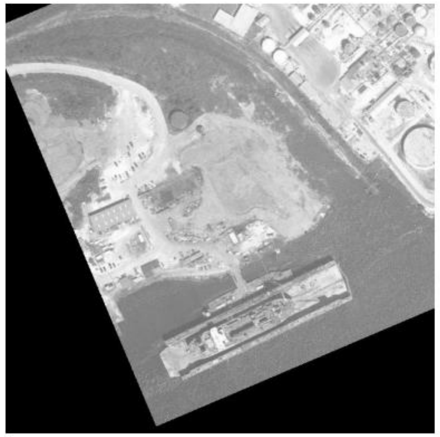


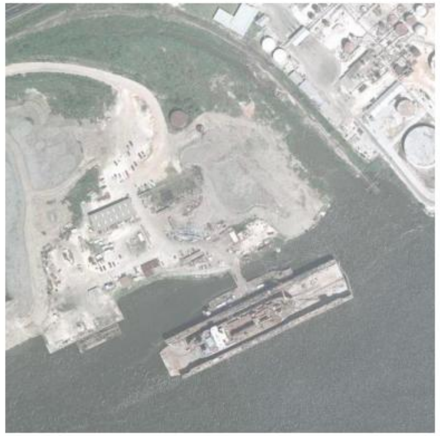
Image $B$

$$u = f_1(x, y)$$
$$v = f_2(x, y)$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



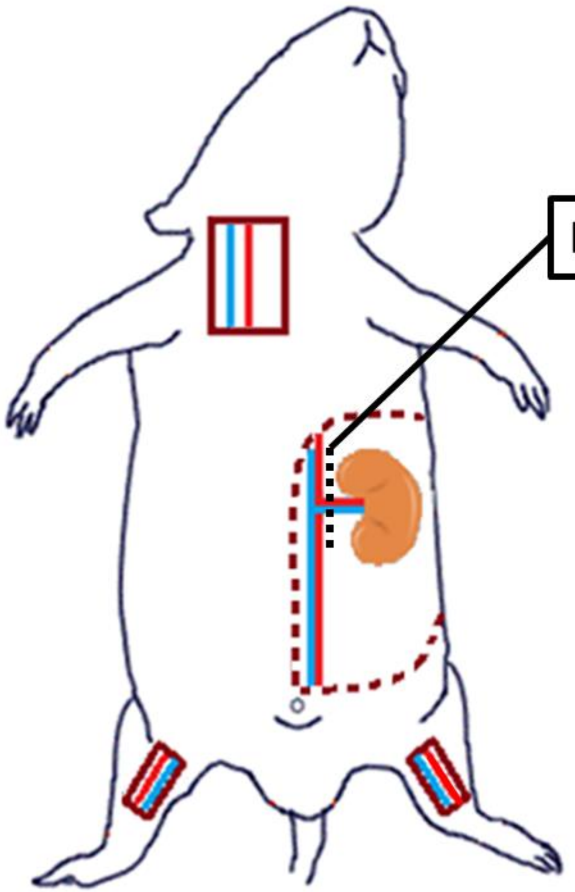
Mapped $A$



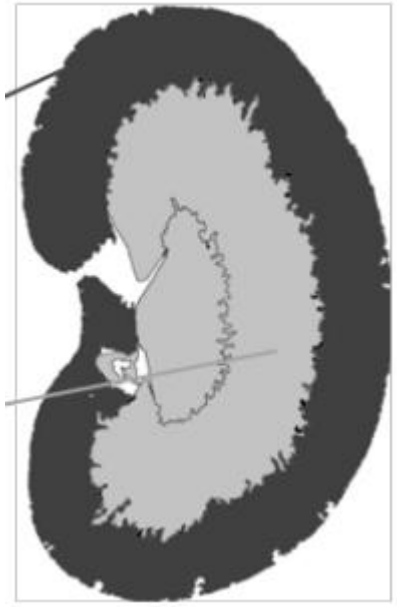
Original $B$

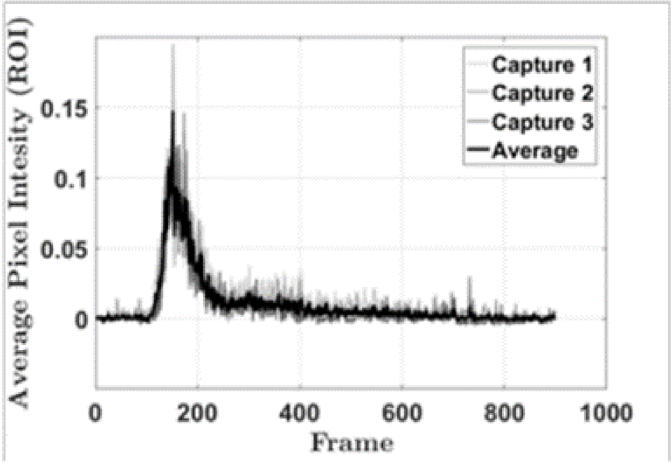
# Model Example 3

- Distribution of Renal Blood Flow
    - Input:
        - Indicator Dilution Curves
    - Model Parameters:
        - Many
    - Output:
        - Cortical blood flow
        - Medullary Blood flow



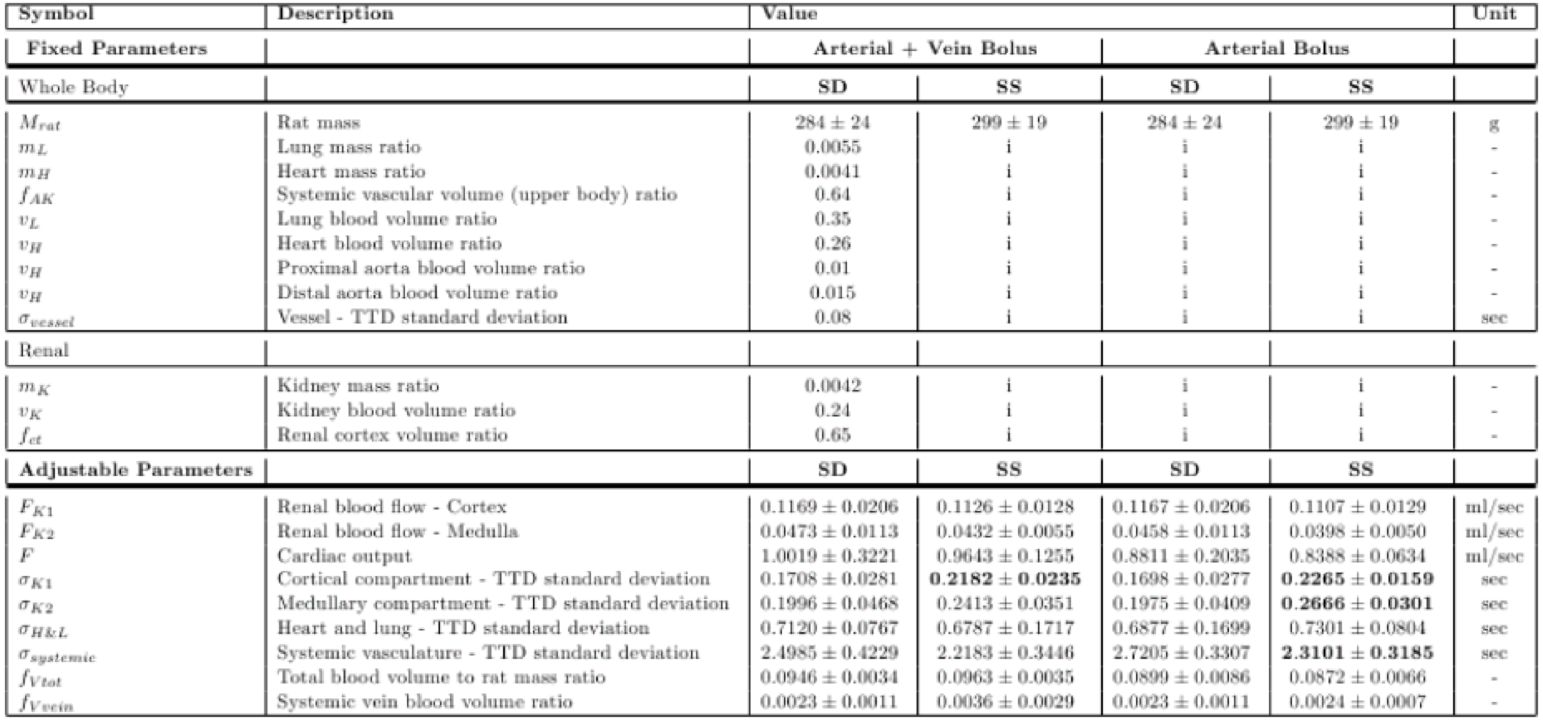
Imaging plane





# Model Example 3

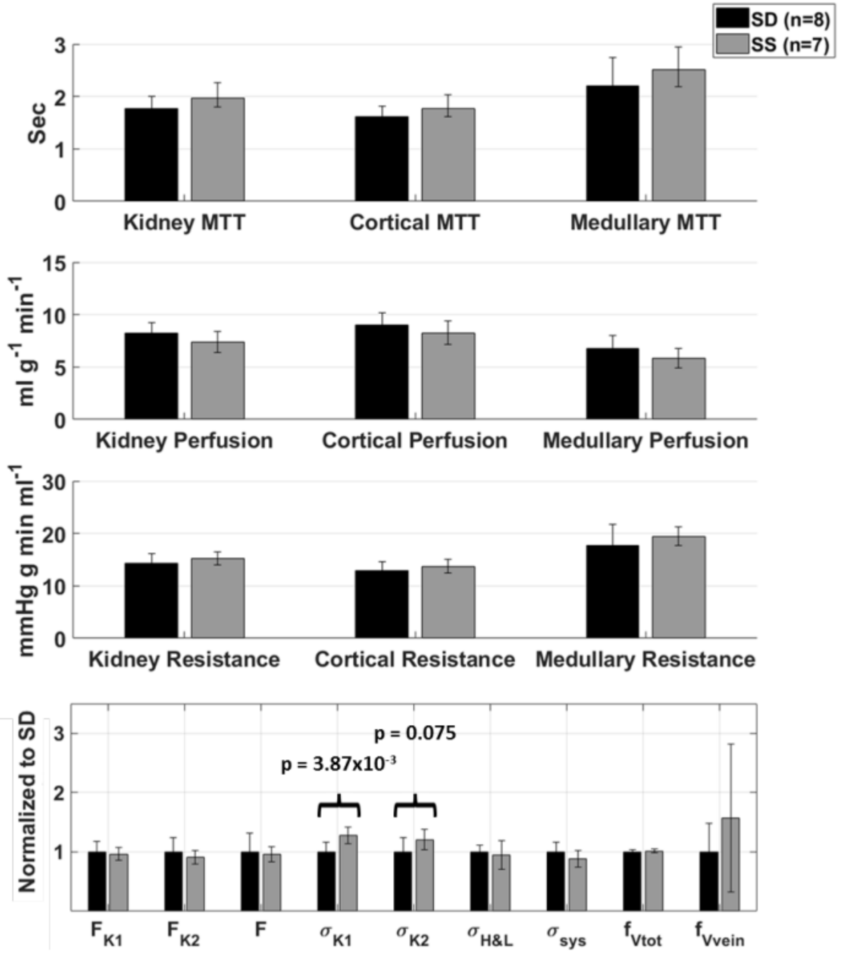
- Distribution of Renal Blood Flow
  - Input:
    - Indicator Dilution Curves
  - Model Parameters:
    - Many
  - Output:
    - Cortical blood flow
    - Medullary Blood flow



| Symbol | Description | Value | | | | Unit |
|---|---|---|---|---|---|---|
| **Fixed Parameters** | | **Arterial + Vein Bolus** | | **Arterial Bolus** | | |
| Whole Body | | **SD** | **SS** | **SD** | **SS** | |
| $M_{rat}$ | Rat mass | $284 \pm 24$ | $299 \pm 19$ | $284 \pm 24$ | $299 \pm 19$ | g |
| $m_L$ | Lung mass ratio | 0.0055 | i | i | i | - |
| $m_H$ | Heart mass ratio | 0.0041 | i | i | i | - |
| $f_{AK}$ | Systemic vascular volume (upper body) ratio | 0.64 | i | i | i | - |
| $v_L$ | Lung blood volume ratio | 0.35 | i | i | i | - |
| $v_H$ | Heart blood volume ratio | 0.26 | i | i | i | - |
| $v_H$ | Proximal aorta blood volume ratio | 0.01 | i | i | i | - |
| $v_H$ | Distal aorta blood volume ratio | 0.015 | i | i | i | - |
| $\sigma_{vessel}$ | Vessel - TTD standard deviation | 0.08 | i | i | i | sec |
| Renal | | | | | | |
| $m_K$ | Kidney mass ratio | 0.0042 | i | i | i | - |
| $v_K$ | Kidney blood volume ratio | 0.24 | i | i | i | - |
| $f_{ct}$ | Renal cortex volume ratio | 0.65 | i | i | i | - |
| **Adjustable Parameters** | | **SD** | **SS** | **SD** | **SS** | |
| $F_{K1}$ | Renal blood flow - Cortex | $0.1169 \pm 0.0206$ | $0.1126 \pm 0.0128$ | $0.1167 \pm 0.0206$ | $0.1107 \pm 0.0129$ | ml/sec |
| $F_{K2}$ | Renal blood flow - Medulla | $0.0473 \pm 0.0113$ | $0.0432 \pm 0.0055$ | $0.0458 \pm 0.0113$ | $0.0398 \pm 0.0050$ | ml/sec |
| $F$ | Cardiac output | $1.0019 \pm 0.3221$ | $0.9643 \pm 0.1255$ | $0.8811 \pm 0.2035$ | $0.8388 \pm 0.0634$ | ml/sec |
| $\sigma_{K1}$ | Cortical compartment - TTD standard deviation | $0.1708 \pm 0.0281$ | $\mathbf{0.2182 \pm 0.0235}$ | $0.1698 \pm 0.0277$ | $\mathbf{0.2265 \pm 0.0159}$ | sec |
| $\sigma_{K2}$ | Medullary compartment - TTD standard deviation | $0.1996 \pm 0.0468$ | $0.2413 \pm 0.0351$ | $0.1975 \pm 0.0409$ | $\mathbf{0.2666 \pm 0.0301}$ | sec |
| $\sigma_{H\&L}$ | Heart and lung - TTD standard deviation | $0.7120 \pm 0.0767$ | $0.6787 \pm 0.1717$ | $0.6877 \pm 0.1699$ | $0.7301 \pm 0.0804$ | sec |
| $\sigma_{systemic}$ | Systemic vasculature - TTD standard deviation | $2.4985 \pm 0.4229$ | $2.2183 \pm 0.3446$ | $2.7205 \pm 0.3307$ | $\mathbf{2.3101 \pm 0.3185}$ | sec |
| $f_{Vtot}$ | Total blood volume to rat mass ratio | $0.0946 \pm 0.0034$ | $0.0963 \pm 0.0035$ | $0.0899 \pm 0.0086$ | $0.0872 \pm 0.0066$ | - |
| $f_{Vvein}$ | Systemic vein blood volume ratio | $0.0023 \pm 0.0011$ | $0.0036 \pm 0.0029$ | $0.0023 \pm 0.0011$ | $0.0024 \pm 0.0007$ | - |

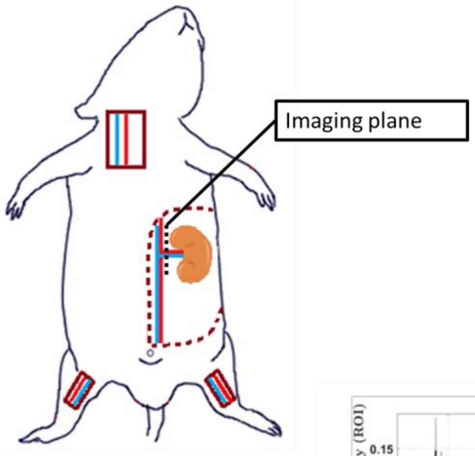
# Model Example 3

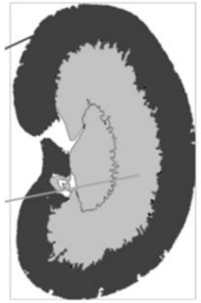
- Distribution of Renal Blood Flow
  - Input:
    - Indicator Dilution Curves
  - Model Parameters:
    - Many
  - Output:
    - Cortical blood flow
    - Medullary Blood flow

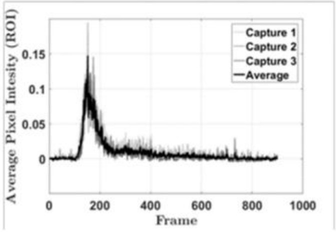


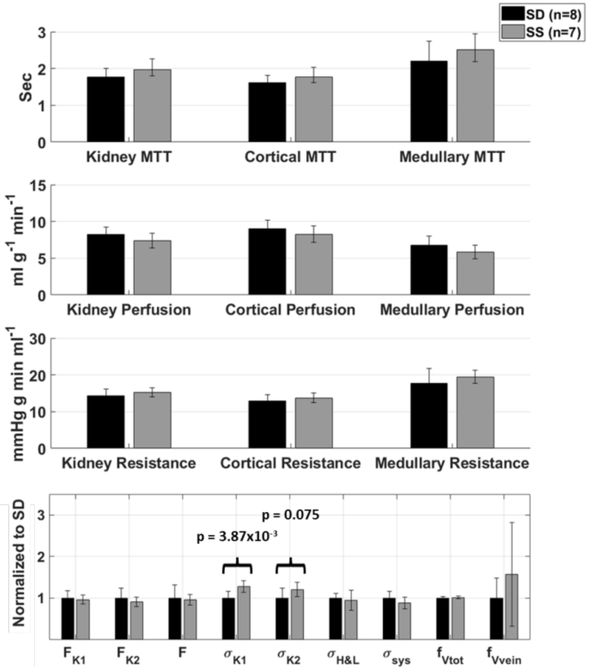
# Model Example 3

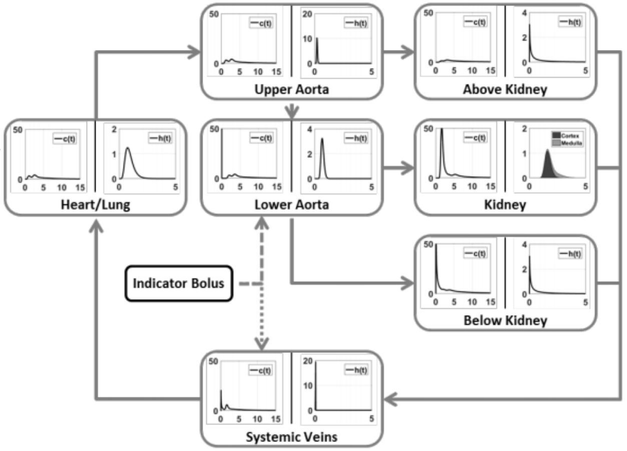
- Distribution of Renal Blood Flow
  - Input:
    - Indicator Dilution Curves
  - Model Parameters:
    - Many
  - Output:
    - Cortical blood flow
    - Medullary Blood flow

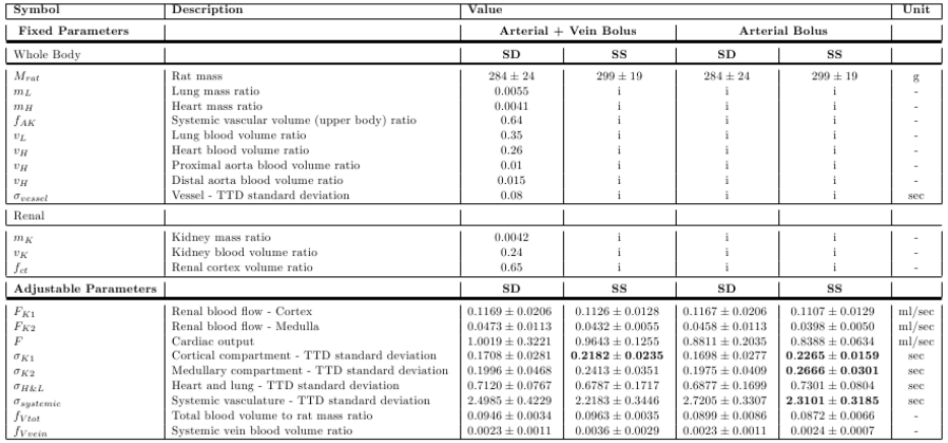








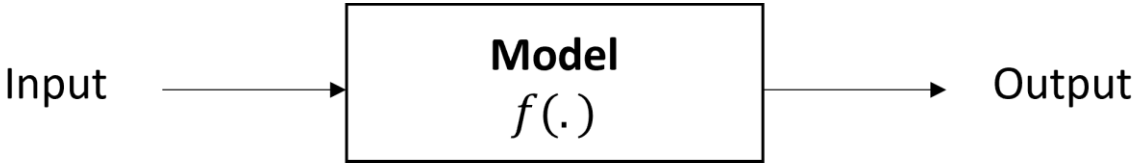




| Symbol | Description | Value | | | | Unit |
|---|---|---|---|---|---|---|
| **Fixed Parameters** | | **Arterial + Vein Bolus** | | **Arterial Bolus** | | |
| Whole Body | | **SD** | **SS** | **SD** | **SS** | |
| $M_{rat}$ | Rat mass | 284 ± 24 | 299 ± 19 | 284 ± 24 | 299 ± 19 | g |
| $m_L$ | Lung mass ratio | 0.0055 | i | i | i | - |
| $m_H$ | Heart mass ratio | 0.0041 | i | i | i | - |
| $f_{AK}$ | Systemic vascular volume (upper body) ratio | 0.64 | i | i | i | - |
| $v_L$ | Lung blood volume ratio | 0.35 | i | i | i | - |
| $v_H$ | Heart blood volume ratio | 0.26 | i | i | i | - |
| $v_H$ | Proximal aorta blood volume ratio | 0.01 | i | i | i | - |
| $v_H$ | Distal aorta blood volume ratio | 0.015 | i | i | i | - |
| $\sigma_{vessel}$ | Vessel - TTD standard deviation | 0.08 | i | i | i | sec |
| Renal | | | | | | |
| $m_K$ | Kidney mass ratio | 0.0042 | i | i | i | - |
| $v_K$ | Kidney blood volume ratio | 0.24 | i | i | i | - |
| $f_{ct}$ | Renal cortex volume ratio | 0.65 | i | i | i | - |
| **Adjustable Parameters** | | **SD** | **SS** | **SD** | **SS** | |
| $F_{K1}$ | Renal blood flow - Cortex | 0.1169 ± 0.0206 | 0.1126 ± 0.0128 | 0.1167 ± 0.0206 | 0.1107 ± 0.0129 | ml/sec |
| $F_{K2}$ | Renal blood flow - Medulla | 0.0473 ± 0.0113 | 0.0432 ± 0.0055 | 0.0458 ± 0.0113 | 0.0398 ± 0.0050 | ml/sec |
| $F$ | Cardiac output | 1.0019 ± 0.3221 | 0.9643 ± 0.1255 | 0.8811 ± 0.2035 | 0.8388 ± 0.0634 | ml/sec |
| $\sigma_{K1}$ | Cortical compartment - TTD standard deviation | 0.1708 ± 0.0281 | **0.2182 ± 0.0235** | 0.1698 ± 0.0277 | **0.2265 ± 0.0159** | sec |
| $\sigma_{K2}$ | Medullary compartment - TTD standard deviation | 0.1996 ± 0.0468 | 0.2413 ± 0.0351 | 0.1975 ± 0.0409 | **0.2666 ± 0.0301** | sec |
| $\sigma_{H\&L}$ | Heart and lung - TTD standard deviation | 0.7120 ± 0.0767 | 0.6787 ± 0.1717 | 0.6877 ± 0.1699 | 0.7301 ± 0.0804 | sec |
| $\sigma_{systemic}$ | Systemic vasculature - TTD standard deviation | 2.4985 ± 0.4229 | 2.2183 ± 0.3446 | 2.7205 ± 0.3307 | **2.3101 ± 0.3185** | sec |
| $f_{Vtot}$ | Total blood volume to rat mass ratio | 0.0946 ± 0.0034 | 0.0963 ± 0.0035 | 0.0899 ± 0.0086 | 0.0872 ± 0.0066 | - |
| $f_{Vvein}$ | Systemic vein blood volume ratio | 0.0023 ± 0.0011 | 0.0036 ± 0.0029 | 0.0023 ± 0.0011 | 0.0024 ± 0.0007 | - |

# Machine Learning Model

# Machine Learning Models – Supervised Learning

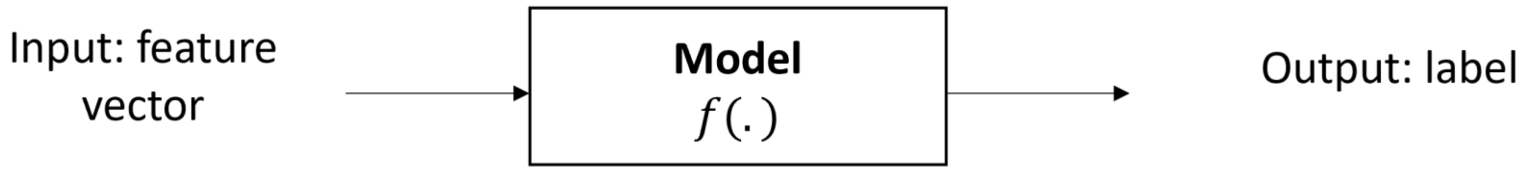


- Supervised machine learning:

Input?

Output?

# Machine Learning Models

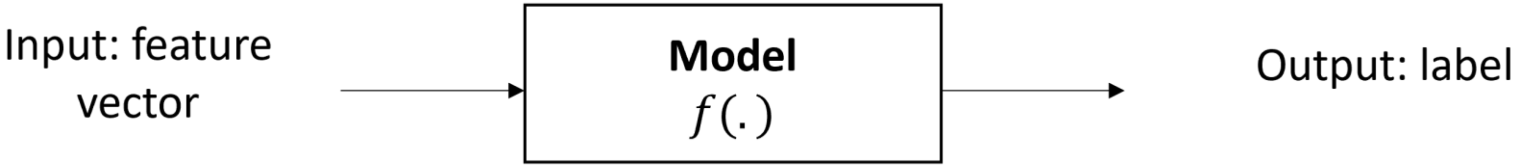


$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

$$y = f(\boldsymbol{x})$$

| | |
|---|---|
| Features of a patient | Presence or not of heart disease |
| Features of a house | Price of a house |
| Features of a flower | Setosa/ Virginica/ Versicolor |
| Features of an MRI image | Presence of cancer cells |

# Machine Learning Models



Input: feature vector

**Model**

$f(.)$

Output: label

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

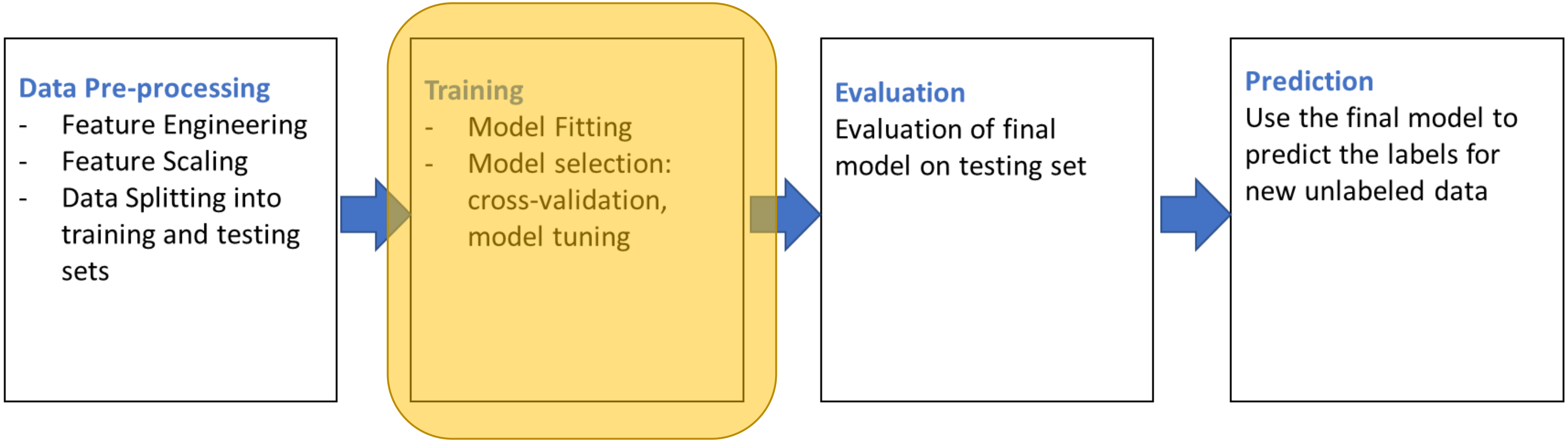
$$y = f(\boldsymbol{x})$$

- In supervised machine learning, we assume that there is an underlying relationship between a feature vector and its corresponding label:
  - $\boldsymbol{x}$ contains the set of features (measures, independent variables, attributes, ..) that represents one object
  - $y$ represent what we want to predict for a given feature vector (label, response, output)

# Machine Learning Models

- How to find this machine learning model?

This is the goal of the training phase.



**Data Pre-processing**

- Feature Engineering
- Feature Scaling
- Data Splitting into training and testing sets

**Training**

- Model Fitting
- Model selection: cross-validation, model tuning

**Evaluation**

Evaluation of final model on testing set

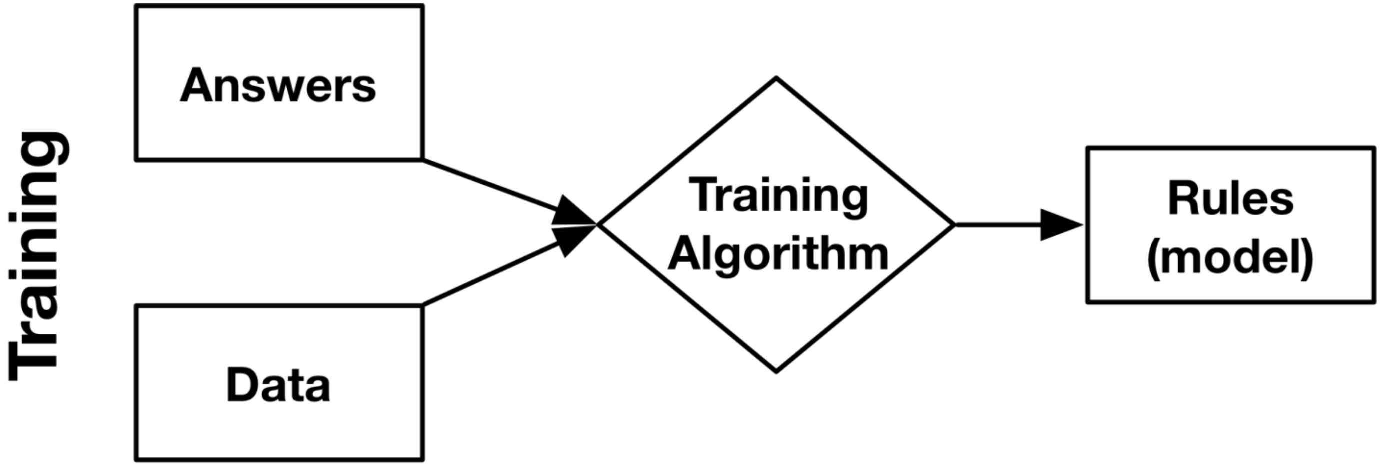
**Prediction**

Use the final model to predict the labels for new unlabeled data

# Training phase – Model fitting (Supervised Learning)

Label vector

$$\boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix}$$



Feature matrix that consists of training data

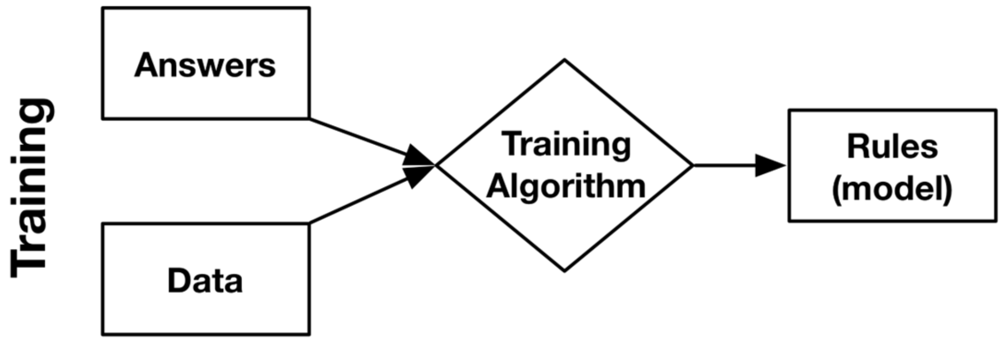
$$X = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_m^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_m^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(n)} & x_2^{(n)} & \dots & x_m^{(n)} \end{bmatrix}$$

# Training phase – Model fitting (Supervised Learning)

- In supervised learning, a training algorithm expects historical data (pairs of features and their labels):
  - Examples of patient features and whether each has heart disease or not
  - Examples of flower features and their corresponding categories
  - Examples of house features and their corresponding price
- The training algorithm uses the historical data to learn the model: it tries to detect on its own the relationship between feature vectors and labels.

Label vector

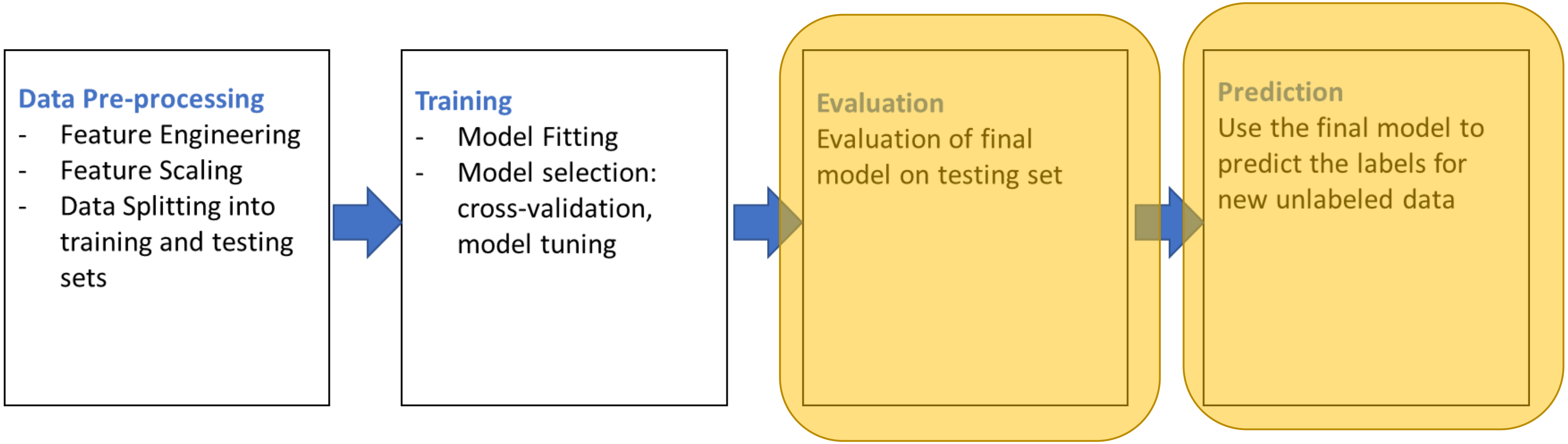
$$\boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix}$$



$$X = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_m^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_m^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(n)} & x_2^{(n)} & \dots & x_m^{(n)} \end{bmatrix}$$

# Machine Learning Models

- Once a final machine learning model is learned. The model is evaluated and then used for future predictions.



**Data Pre-processing**

- Feature Engineering
- Feature Scaling
- Data Splitting into training and testing sets

**Training**

- Model Fitting
- Model selection: cross-validation, model tuning

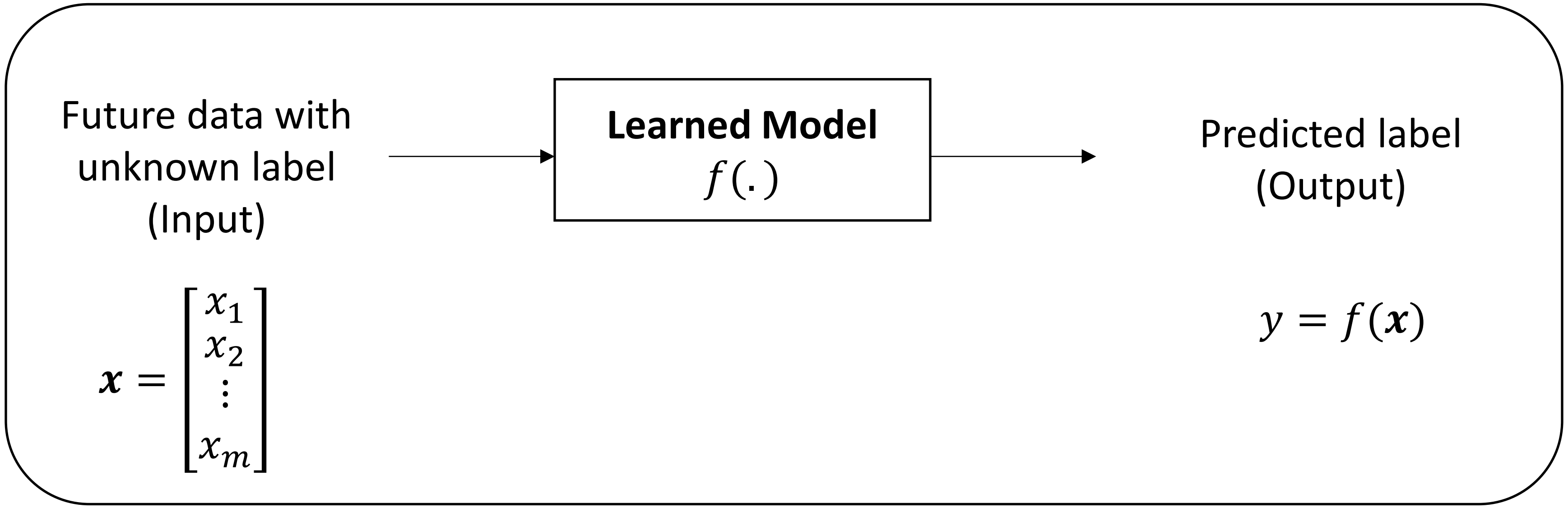
**Evaluation**

Evaluation of final model on testing set

**Prediction**

Use the final model to predict the labels for new unlabeled data

# Using Machine Learning Models



Future data with unknown label (Input)

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

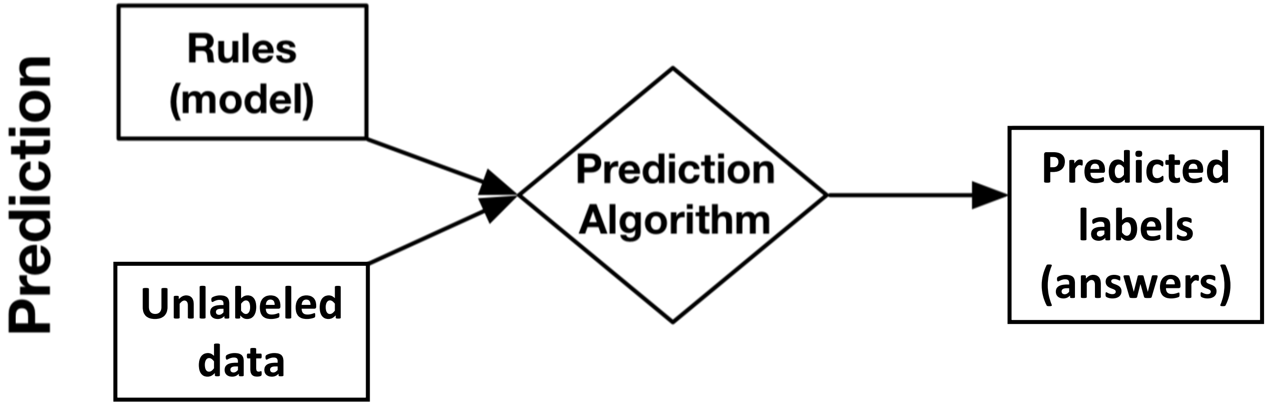
**Learned Model**

$f(.)$

Predicted label (Output)

$$y = f(\boldsymbol{x})$$

# Using Machine Learning Models



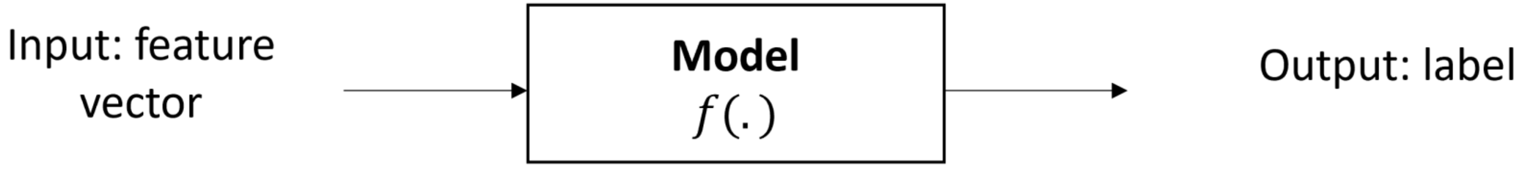
# Machine Learning Models

- Machine learning models consist of linear and non-linear models.

- We will now discuss linear models, and what a linear model means for regression, and what it means for classification. How to geometrically interpret them, and what does training linear models mean.

# Linear model

# Linear Models - Description



Input: feature vector

**Model**
$f(.)$

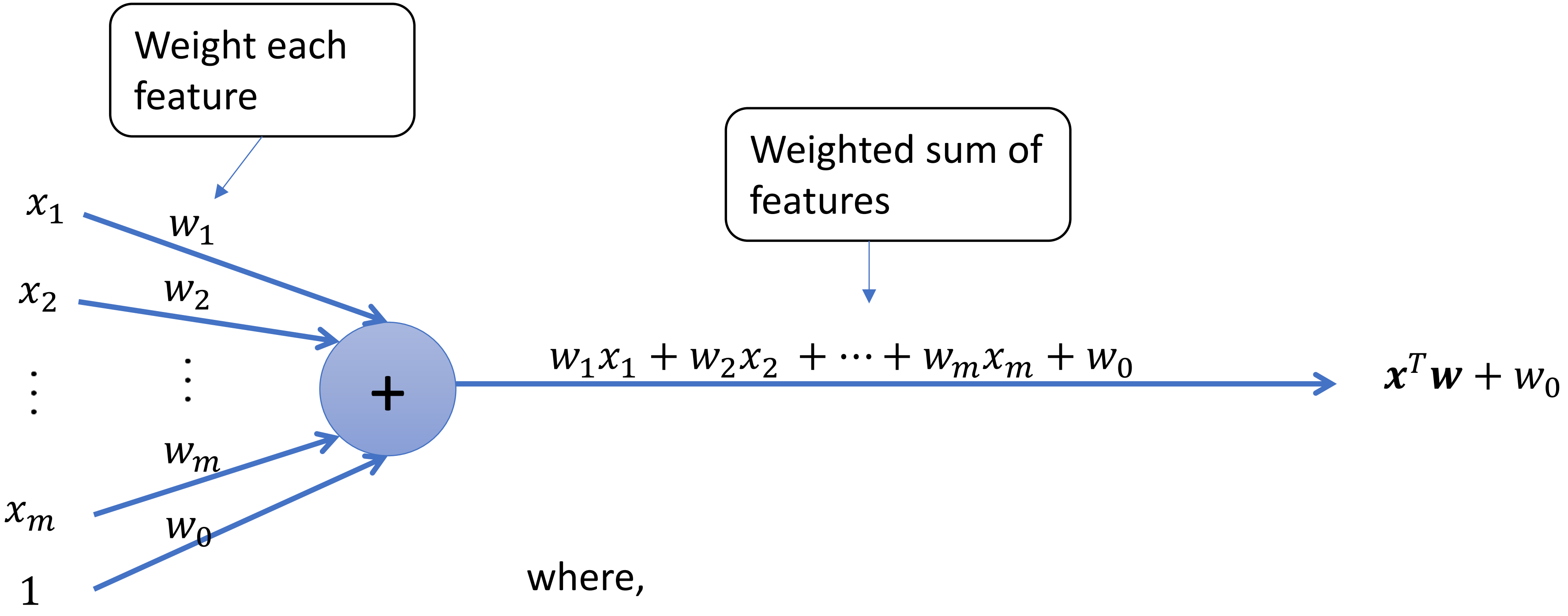
Output: label

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

$$y = f(\boldsymbol{x})$$

What assumptions do linear model make about the form of f(.)?

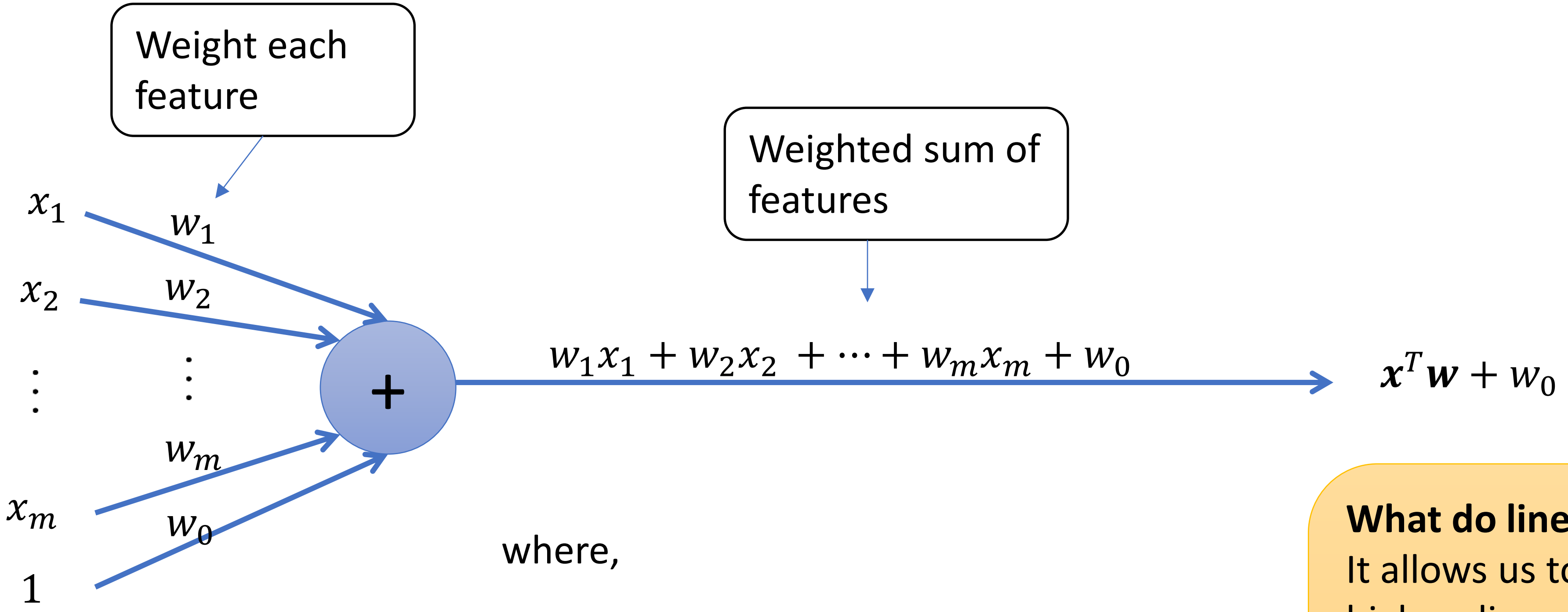
# Linear Models - Description



where,

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \quad \text{and} \quad \boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix}$$

# Linear Models - Description


$x_1$
$w_1$
$x_2$
$w_2$
$\vdots$
$\vdots$
$w_m$
$x_m$
$w_0$
$1$
$+$
$w_1x_1 + w_2x_2 \ + \cdots + w_mx_m + w_0$
$\boldsymbol{x}^T\boldsymbol{w} + w_0$

Weight each feature

Weighted sum of features

where,

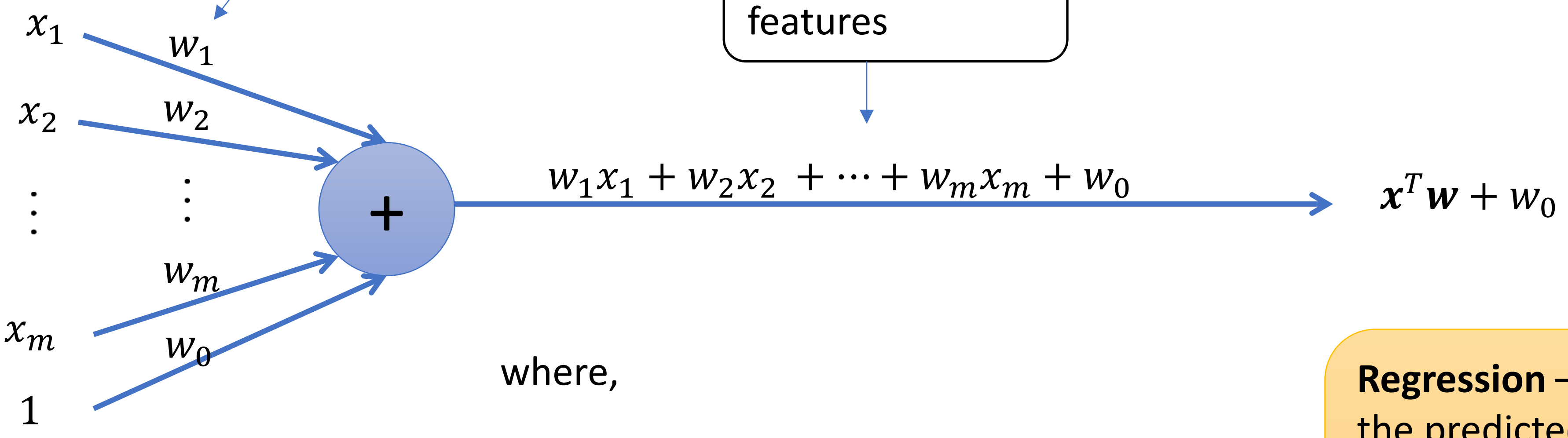
$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \quad \text{and} \quad \boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix}$$

**What do linear models do?**
It allows us to convert from a higher dimensional  space to a scalar value.
It compresses the information contained in each feature in one scalar.

# Linear Models - Description

Weight each feature

Weighted sum of features

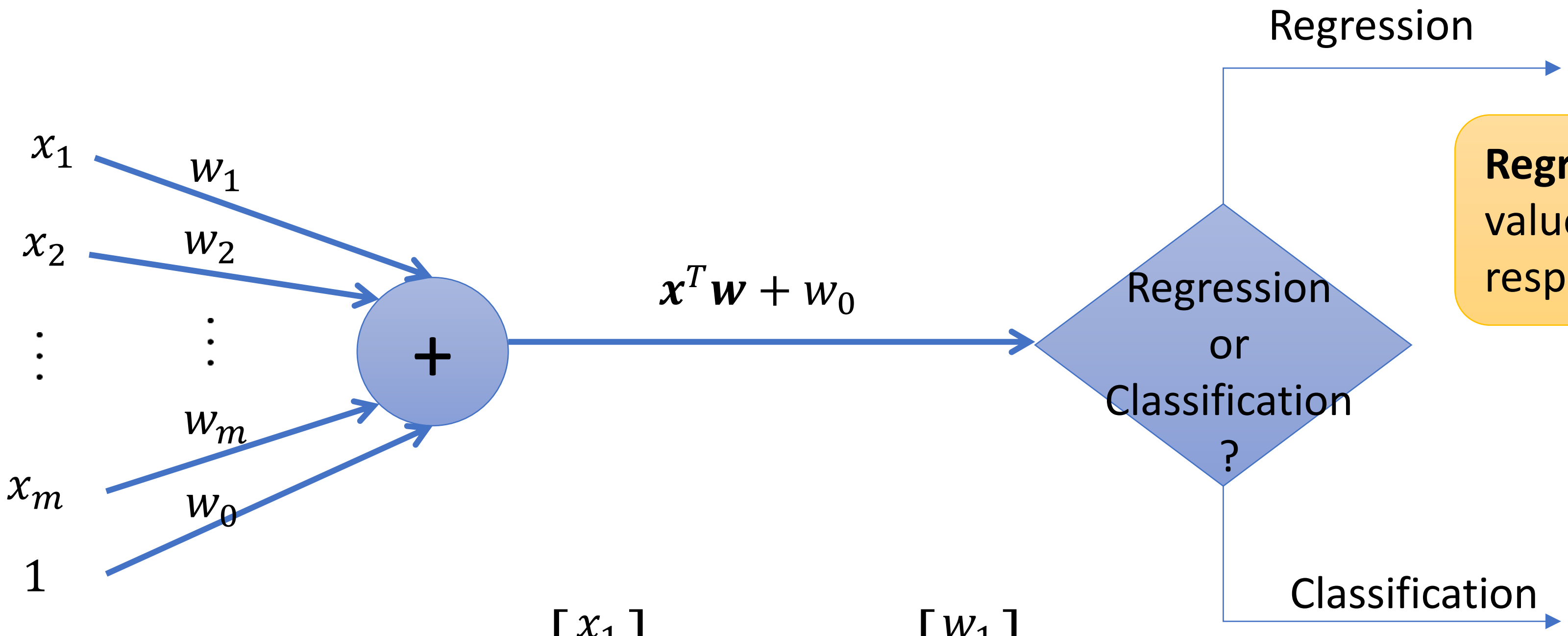

$x_1$
$x_2$
$\vdots$
$x_m$
$1$
$w_1$
$w_2$
$\vdots$
$w_m$
$w_0$
+
$w_1x_1 + w_2x_2 \ + \cdots + w_mx_m + w_0$
$\boldsymbol{x}^T\boldsymbol{w} + w_0$

where,

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \quad \text{and} \quad \boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix}$$

**Regression** – the scalar value is the predicted response. **Classification** - a threshold is applied to this scalar value to predict a class.

# Regression and Classification Linear model



$$y = \boldsymbol{x}^T\boldsymbol{w} + w_0$$

**Regression** – the scalar value is the predicted response (y).

$$\text{y} = \begin{cases} 1 \text{ if } \ \ \boldsymbol{x}^T\boldsymbol{w} + w_0 \geq 0 \\ 0 \text{ if } \ \ \boldsymbol{x}^T\boldsymbol{w} + w_0 \ < 0 \end{cases}$$

**Classification** - a threshold is applied to this scalar value to predict a class.

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \quad \text{and} \quad \boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix}$$

# How do we find the weights of the linear model?

- We find them by training/fitting the linear model on the training data set: the weights are the **parameters** of the linear models.
- What do we mean by model’s parameters?
  - The parameters of a model are the building blocks of a model.
  - The functional form f(.) of a model is defined through the model’s parameters.
  - If we know the parameters of a model, we know everything about it.

# Model Components

- **Parameters:** $\boldsymbol{w} = (w_1, w_2, \cdots, w_m)$ and $w_0$
  - The heart and soul of the linear model
  - Allows the model to adapt to the current problem

- The goal of the training phase is to learn the weights from the training data: pairs of data features and labels.

- We say that the weights summarize the information that is in the training data.

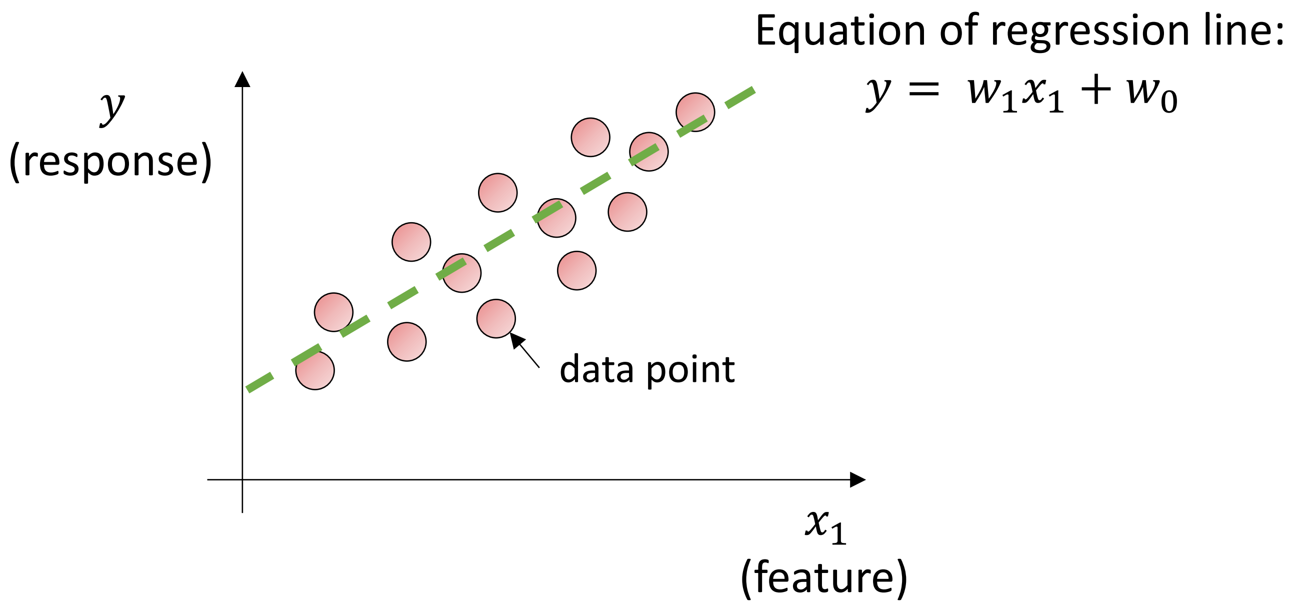
# Linear Regression

# Simple Linear Regression

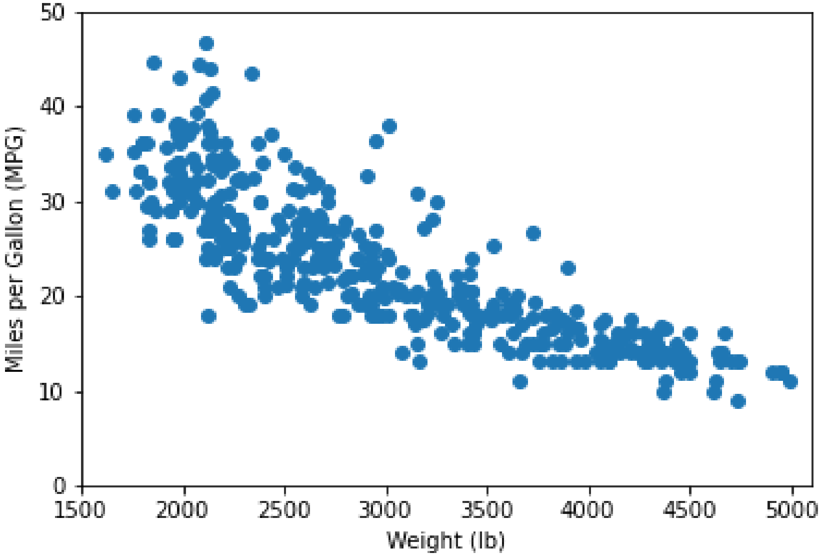
- Simple Linear regression (one feature):

$$y = w_1 x_1 + w_0$$

- We’re interested in predicting $y$ from $x_1$.
- Geometric Interpretation: a simple linear regression model fits the observed training data with a line.



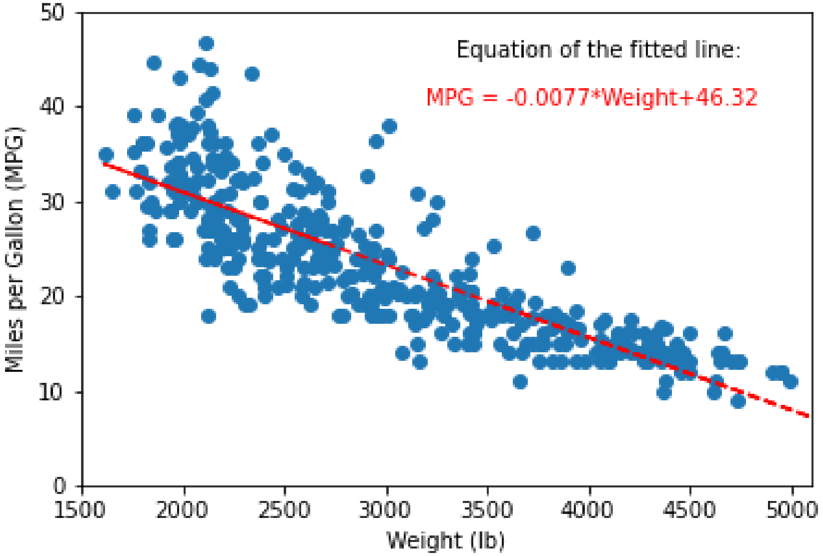
# Simple Linear Regression



Can we predict car fuel consumption in miles per gallon (MPG) from the car's weight?

- We are given a dataset of the weights of several cars and their corresponding MPG.

## Training Phase

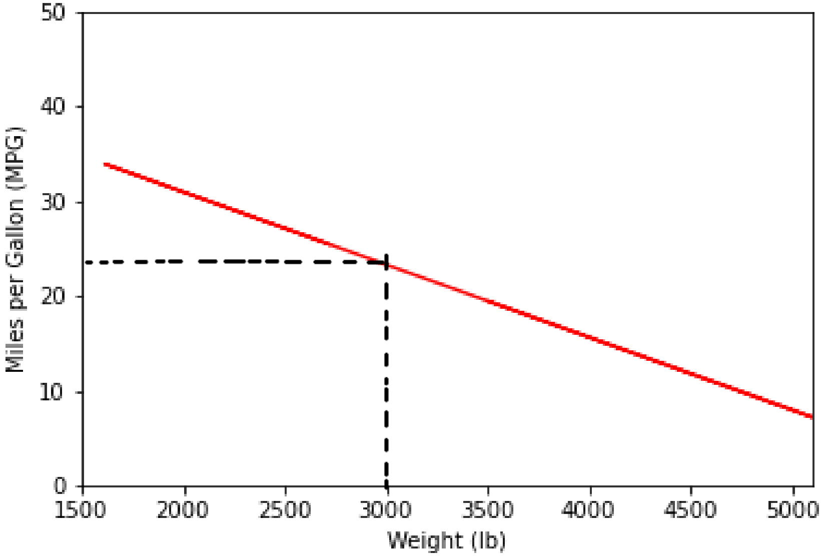


Equation of the fitted line:

MPG = -0.0077*Weight+46.32

During the training phase, the training algorithm tries to find $w_1$ and $w_0$ that best approximate the observed relationship.

## Prediction Phase



Now the line can be used to predict the MPG of a car given its weight.

Predict MPG for car with weight: 3000 lbs.

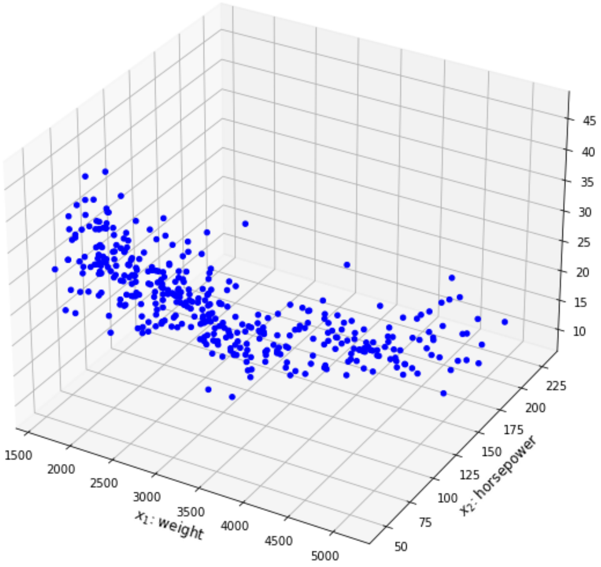
# Multiple Linear Regression

- Multiple Linear regression (more than one feature):

$$y = w_1 x_1 + w_2 x_2 + w_0$$

- We're interested in predicting $y$ from $x_1$ and $x_2$.
- Geometric Interpretation: a multiple linear regression model fits the observed training data (with two features) with a plane.

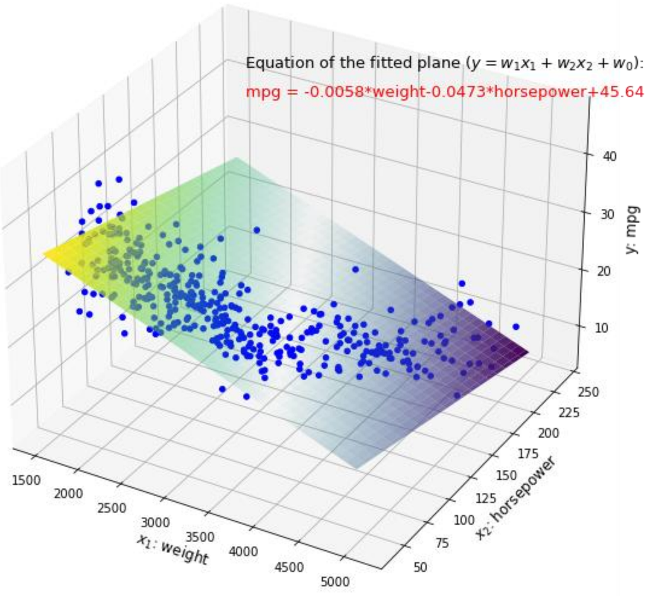
# Multiple Linear Regression



Can we predict car fuel consumption in miles per gallon (MPG) from the car's weight and its horsepower?

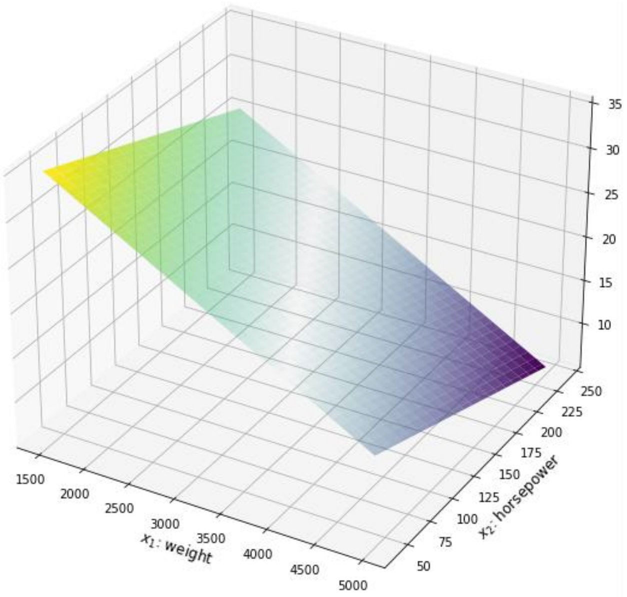
- We are given a dataset of the weights and horsepower of several cars and their corresponding MPG.

**Training Phase**



During the training phase, the training algorithm tries to find $w_1, w_2$ and $w_0$ that best approximate the observed relationship.

**Prediction Phase**



Now the plane can be used to predict the MPG of a car given its weight and horsepower

# Linear Classification (Binary Classification)

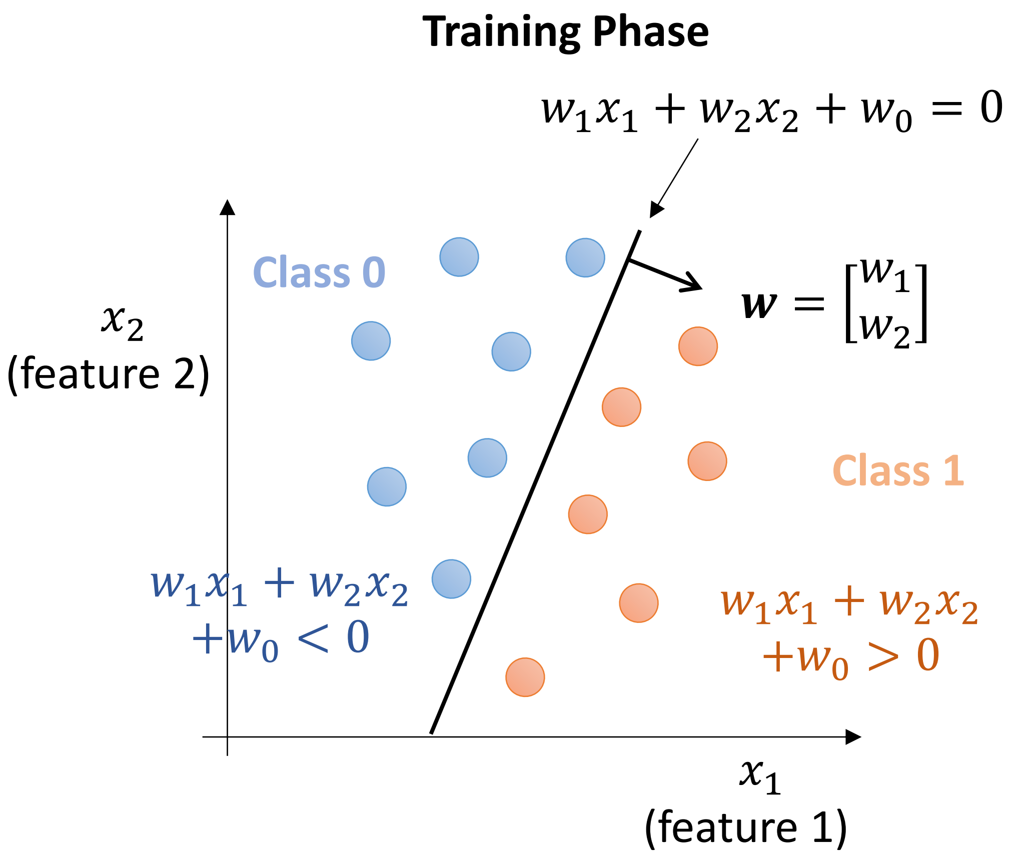
# Geometric Interpretation of Linear Classifiers

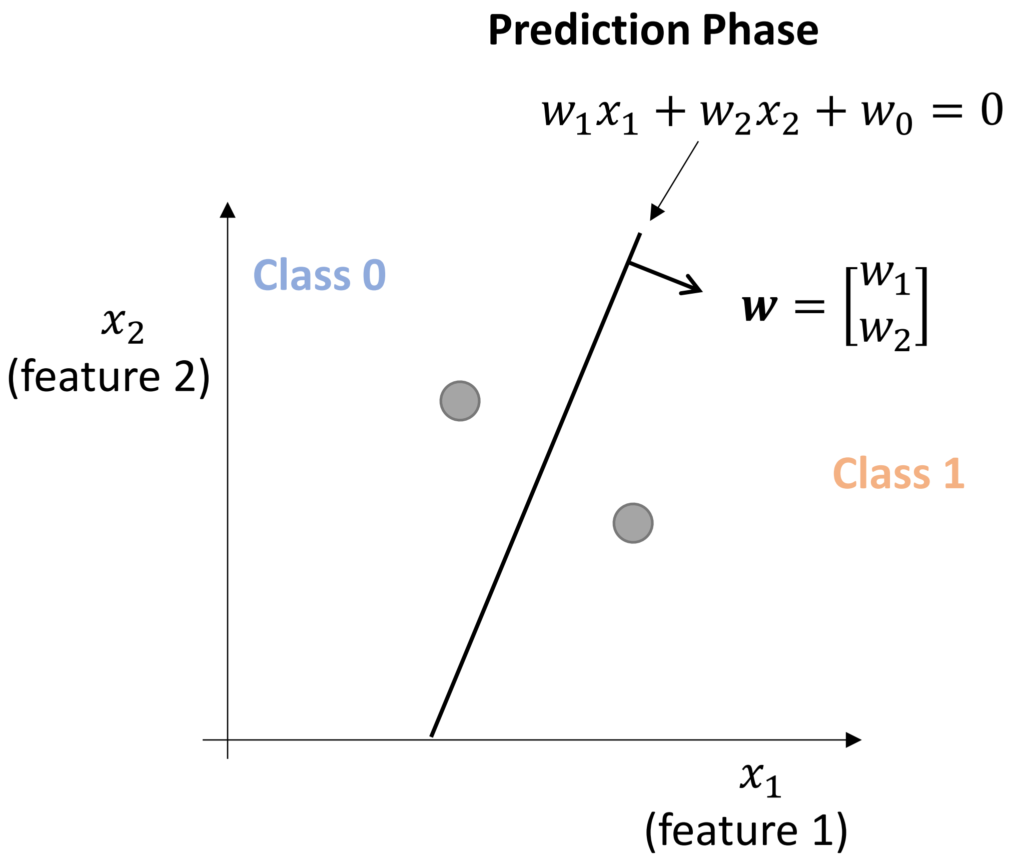
- Training a linear classifier models can be understood as finding the weights such as:

$$\mathrm{y} = \begin{cases} 1 \text{ if } & \boldsymbol{x}^T\boldsymbol{w} + w_0 \geq 0 \\ 0 \text{ if } & \boldsymbol{x}^T\boldsymbol{w} + w_0 \ < 0 \end{cases}$$

- This means that a linear model finds the hyperplane: $\boldsymbol{x}^T\boldsymbol{w} + w_0 = 0$ that divides the feature space (space of all possible feature vectors) into two half-spaces (two decision regions).

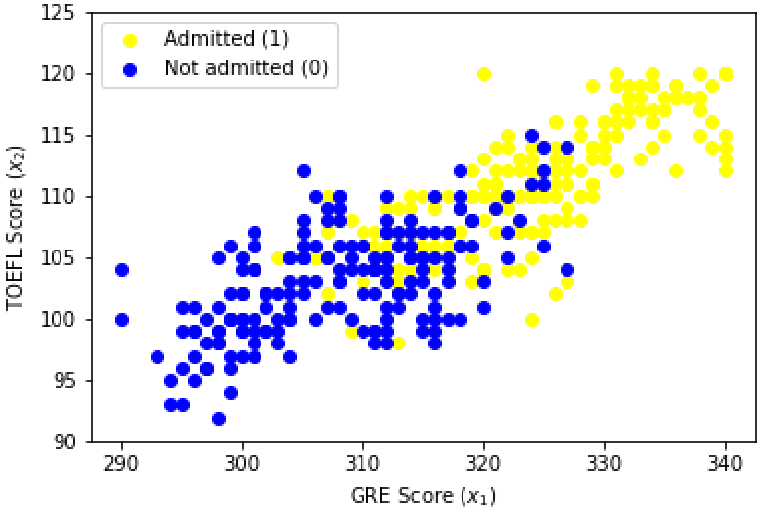
# Linear Classifier – Two-dimensional feature space





Prediction Phase
$w_1x_1 + w_2x_2 + w_0 = 0$
Class 0
$\boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$
$x_2$
(feature 2)
Class 1
$x_1$
(feature 1)

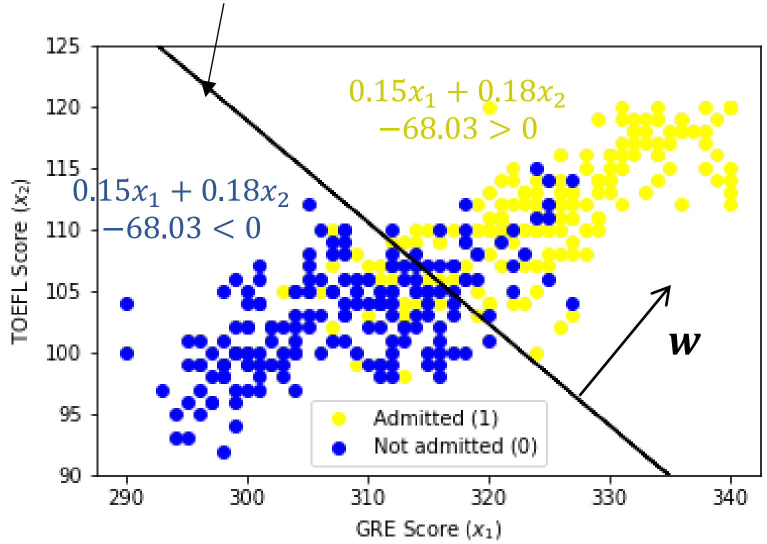
# Example



- GRE scores and TOEFL scores of students who applied to graduated school and whether they were admitted or not.
- Given this data, we're interested in predicting if a student will be admitted or not based on their GRE and TOEFL scores.

## Training phase

$0.15x_1 + 0.18x_2 - 68.03 = 0$

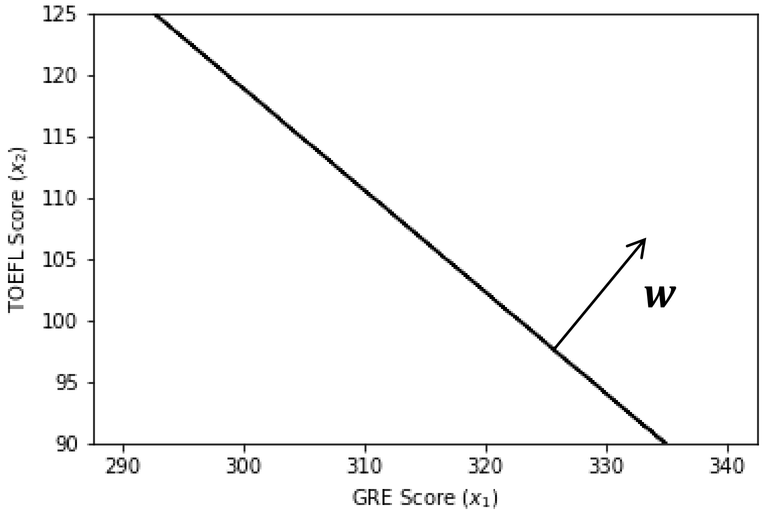


$0.15x_1 + 0.18x_2$
$-68.03 > 0$

$0.15x_1 + 0.18x_2$
$-68.03 < 0$

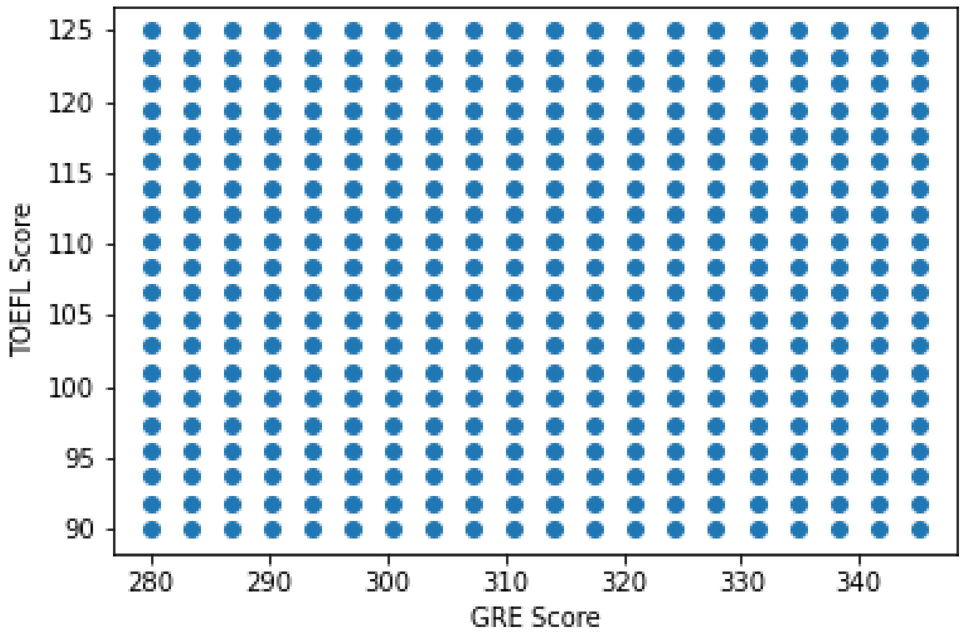
During the training phase, the training algorithm tries to find $w_1, w_2$ and $w_0$ of the line that best separate the two classes.

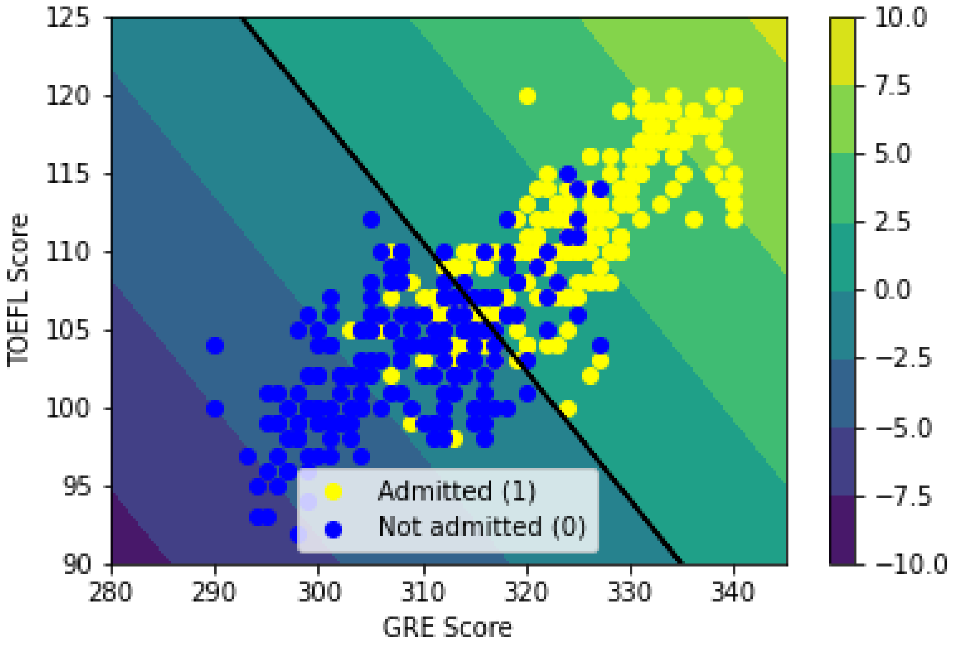
## Prediction phase



Now the line can be used to predict if a student will be admitted based on their two scores.

# Example



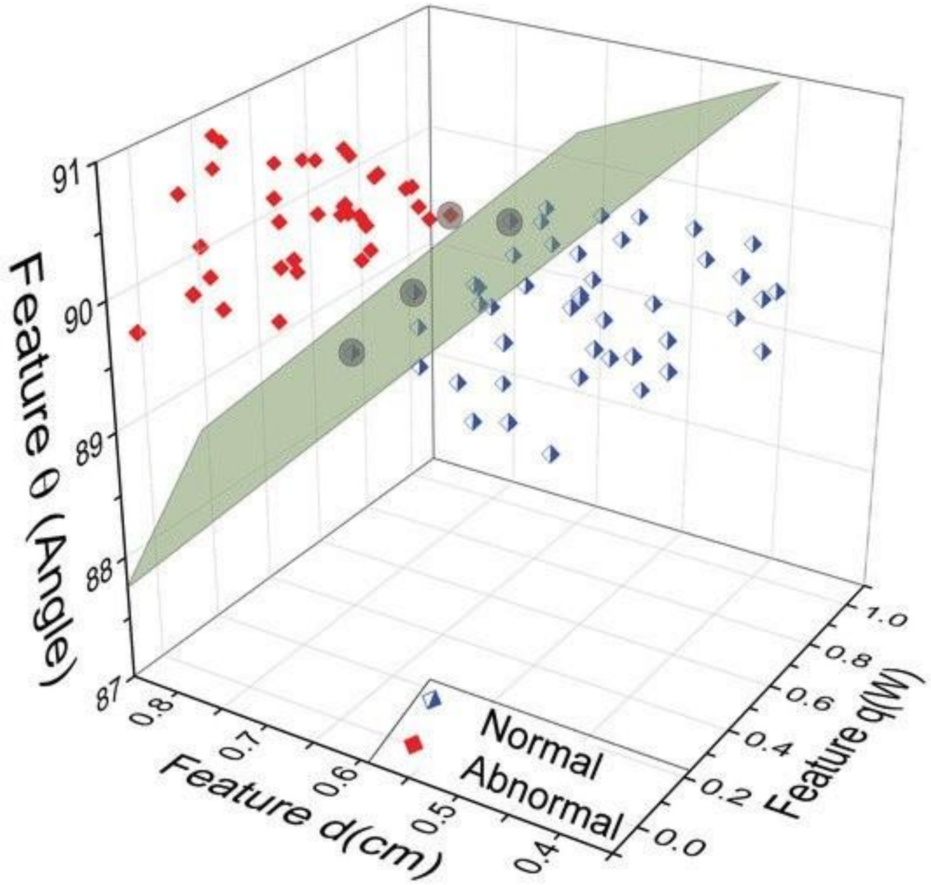


- The colors shown on the right figure represent the actual values (levels) of the weighted sum in the feature space.
- This plot is generated by evaluating the weighted sum for each point shown on the grid of the left figure.
- We will learn later that with logistic regression, the higher (lower) this value is, the more the model is certain that the feature vector corresponds to class 1 (0).

# Linear Classifier – Example in 3D feature space

- The figure shows three physiological features (depth, intensity, radius) that  were extracted from breast thermograms to classify normal and abnormal breast thermograms

Image Source



# Training Algorithm for Linear Models

# Training Linear models

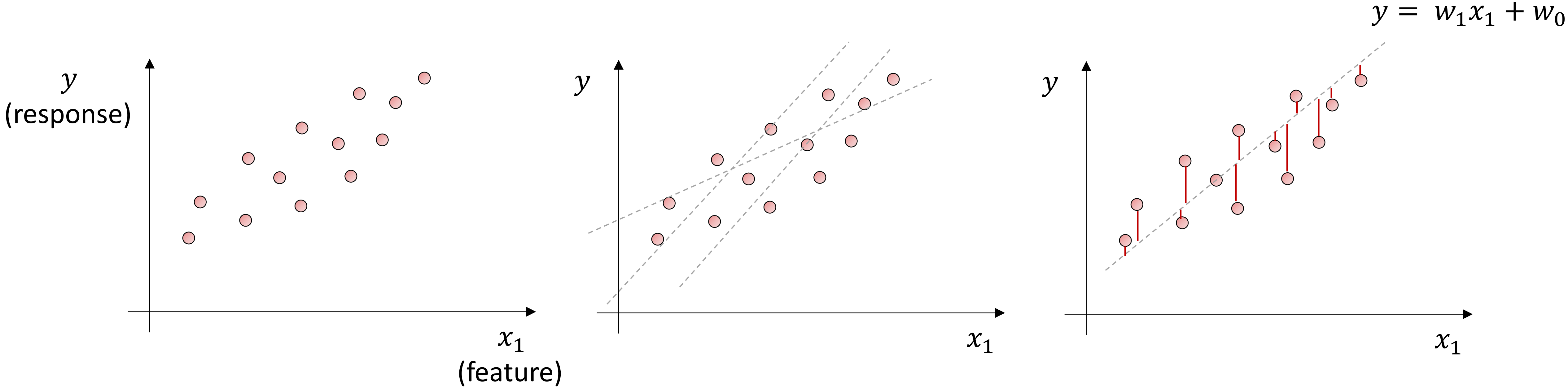
- For both linear regression and linear classification (logistic regression, linear SVM), the problem of finding the weights of the model given the training set is formulated as an optimization problem.

- In the second half of the course, we will look at how each linear model formulates the problem of model fitting as an optimization problem and how we can solve such optimization problem.

# Training Linear Regression

**Simple Linear Regression**

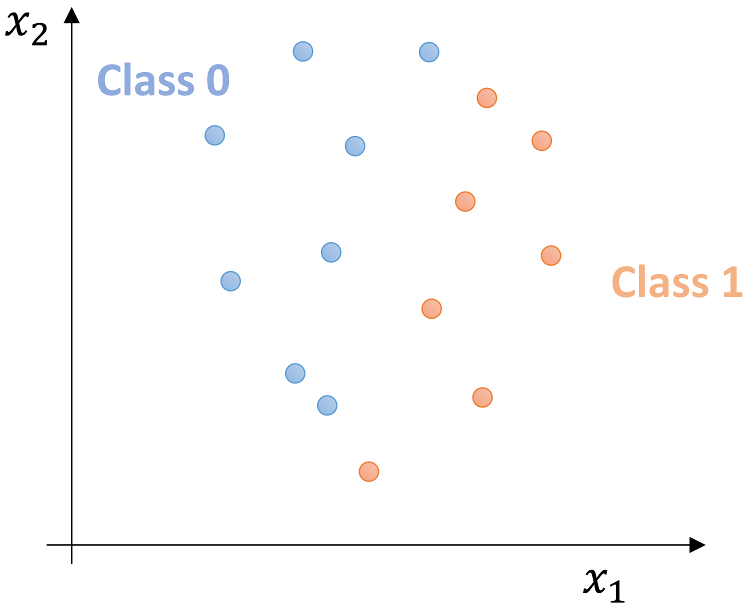
**Training Linear Regression**



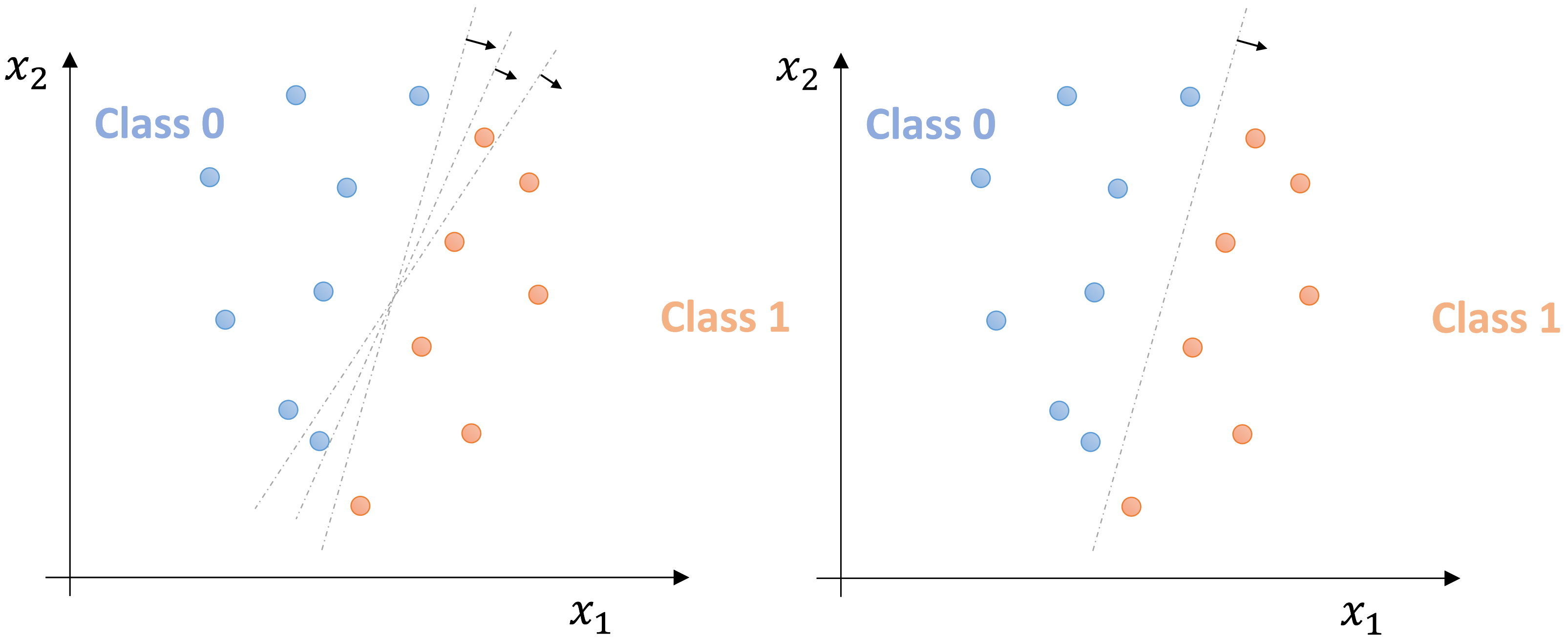
The fitted line is chosen by minimizing the sum of the squared vertical distances between each observation and the line.

# Training a linear classifier

**Linear Classification**



**Training Linear Classification**



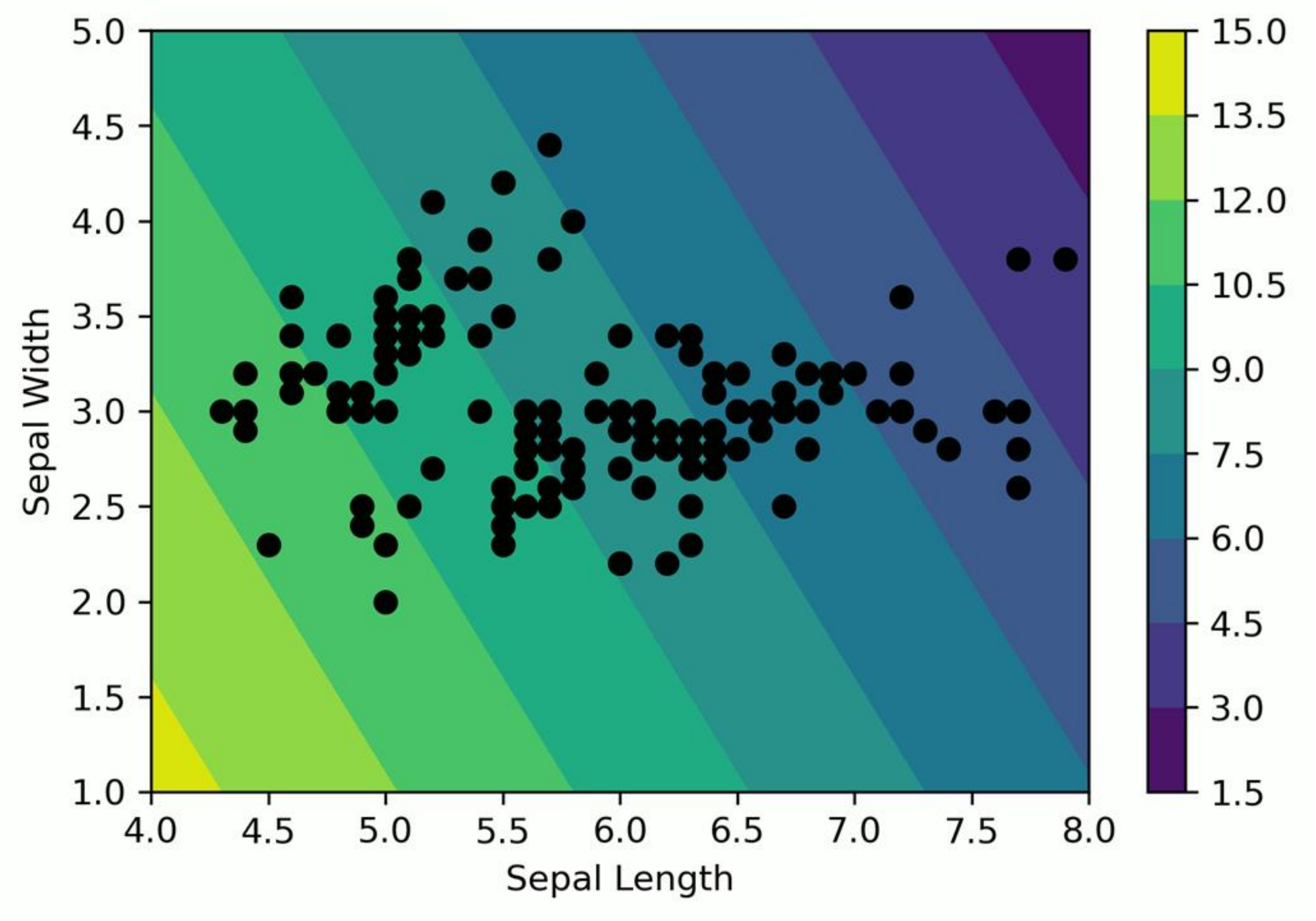
- The separating hyperplane is chosen by minimizing the overall miss-classification.

# Changing Model Parameters

Let $g(x) = \boldsymbol{x}^T\boldsymbol{w} + w_0$

$$\boldsymbol{g(x) = 0.75x_1 - x_2 - 0.9}$$

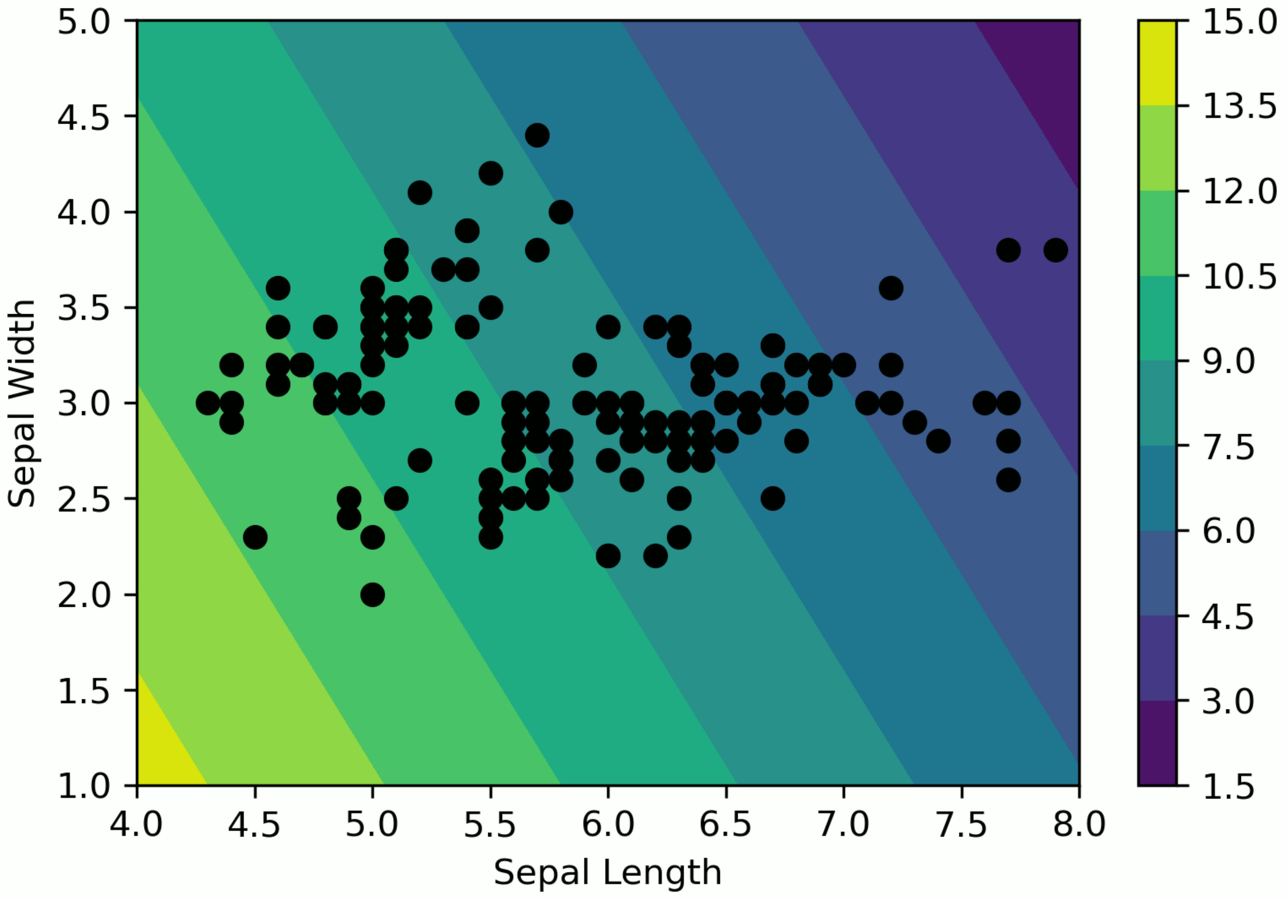
$$w = \begin{bmatrix} 0.75 \\ -1 \end{bmatrix} \leftarrow \textbf{Sweep } \boldsymbol{w_1}$$



# Changing Model Parameters

$$g(x) = \boldsymbol{x}^T \boldsymbol{w} + w_0$$

$$w = \begin{bmatrix} 0.75 \\ -1 \end{bmatrix} \leftarrow$$ **Sweep $w_1$**



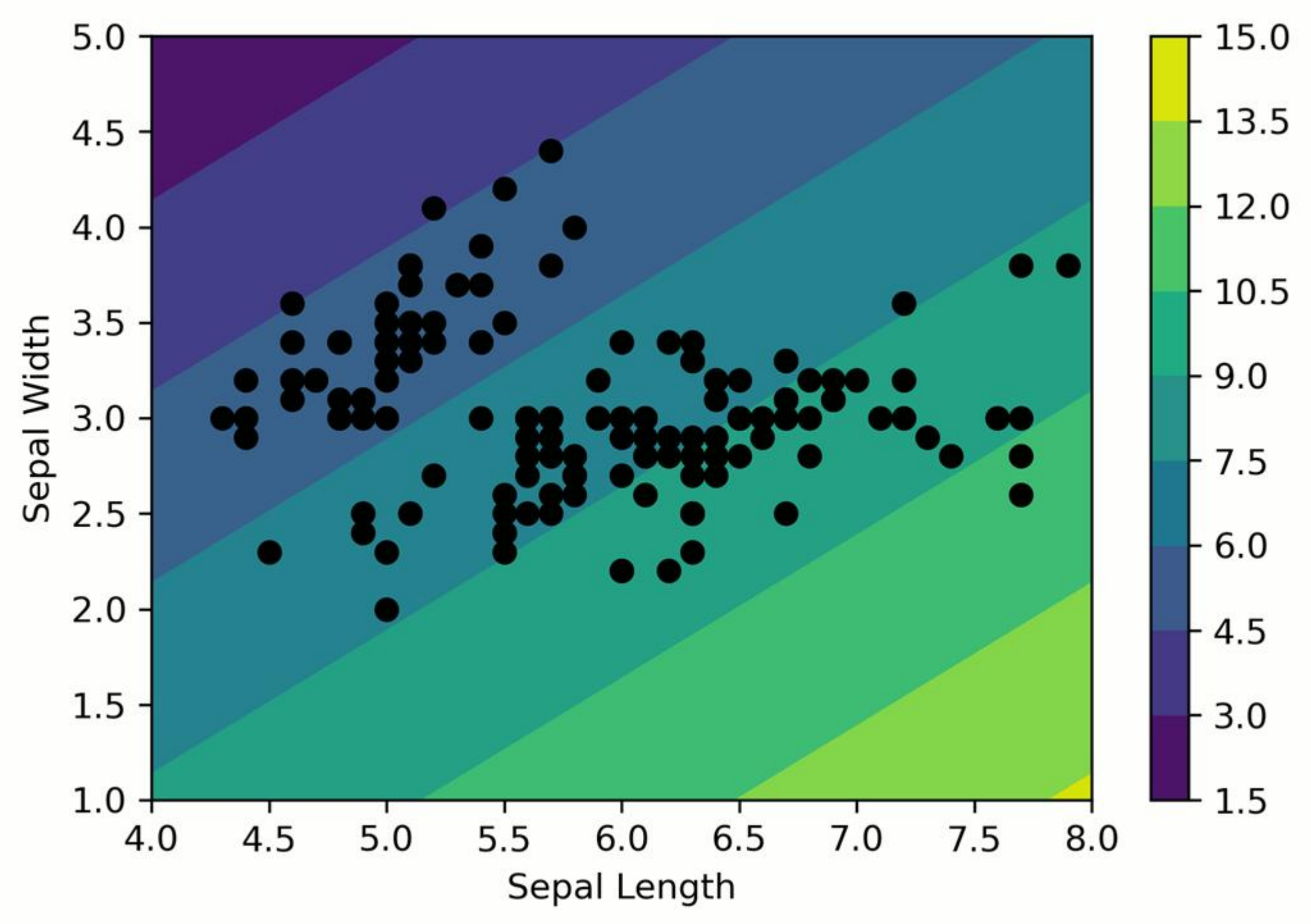
# Changing Model Parameters

$$g(x) = \boldsymbol{x}^T \boldsymbol{w} + w_0$$

$$w = \begin{bmatrix} 0.75 \\ -1 \end{bmatrix}$$

← **Sweep $w_1$**

**Sweep $w_0$**

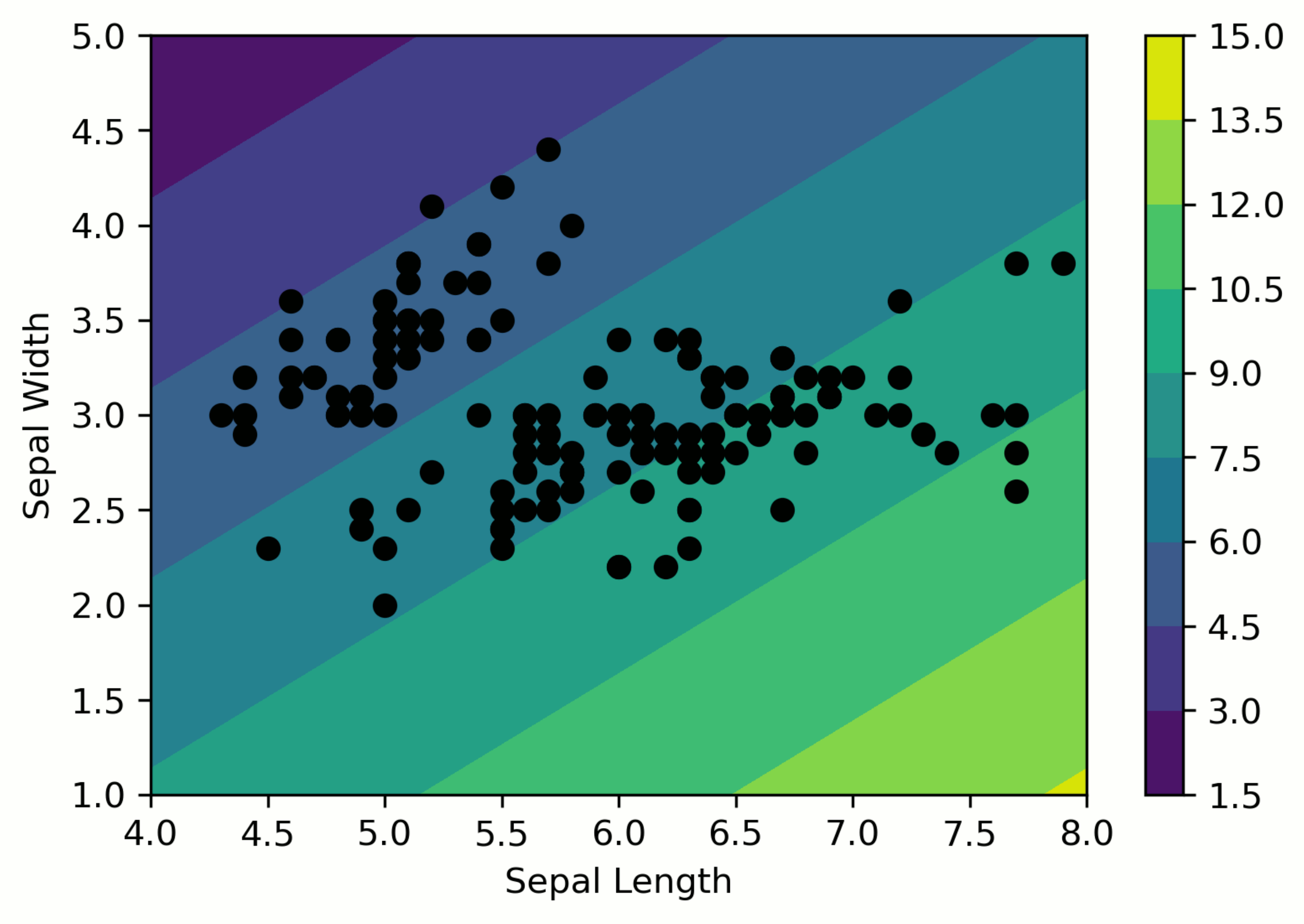


# Changing Model Parameters

$$g(x) = \boldsymbol{x}^T \boldsymbol{w} + w_0$$

$$w = \begin{bmatrix} 0.75 \\ -1 \end{bmatrix}$$

**Sweep $w_0$**



# Linear Models

## Advantages

- Simple model
- When used for prediction, they are computationally fast. Since they summarize data with a finite set of weights, storing trained linear models only requires storing the weights.
- Linear model are interpretable: the weights helps us understand the contribution of each feature to the overall prediction.

## Disadvantages

Linear models has limitations. They can be too simple:

- What if classes are not linearly separable ? (classification)
- What if features are not linearly related to the response?