Aprendizado de Máquina e Reconhecimento de Padrões

Polynomial Logistic Regression

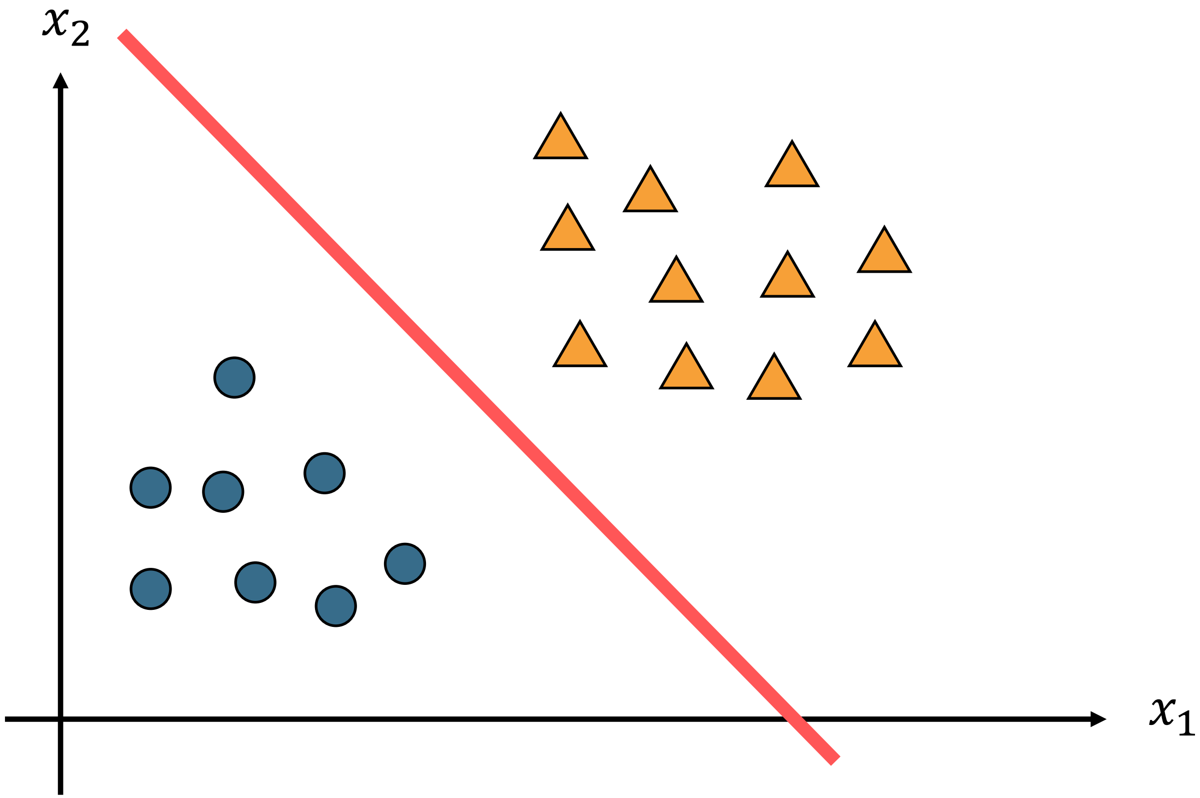
Prof. Dr. Samuel Martins (Samuka)
samuel.martins@ifsp.edu.br

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
São Paulo
Campus Campinas

Linear and Nonlinear Separability of Classes

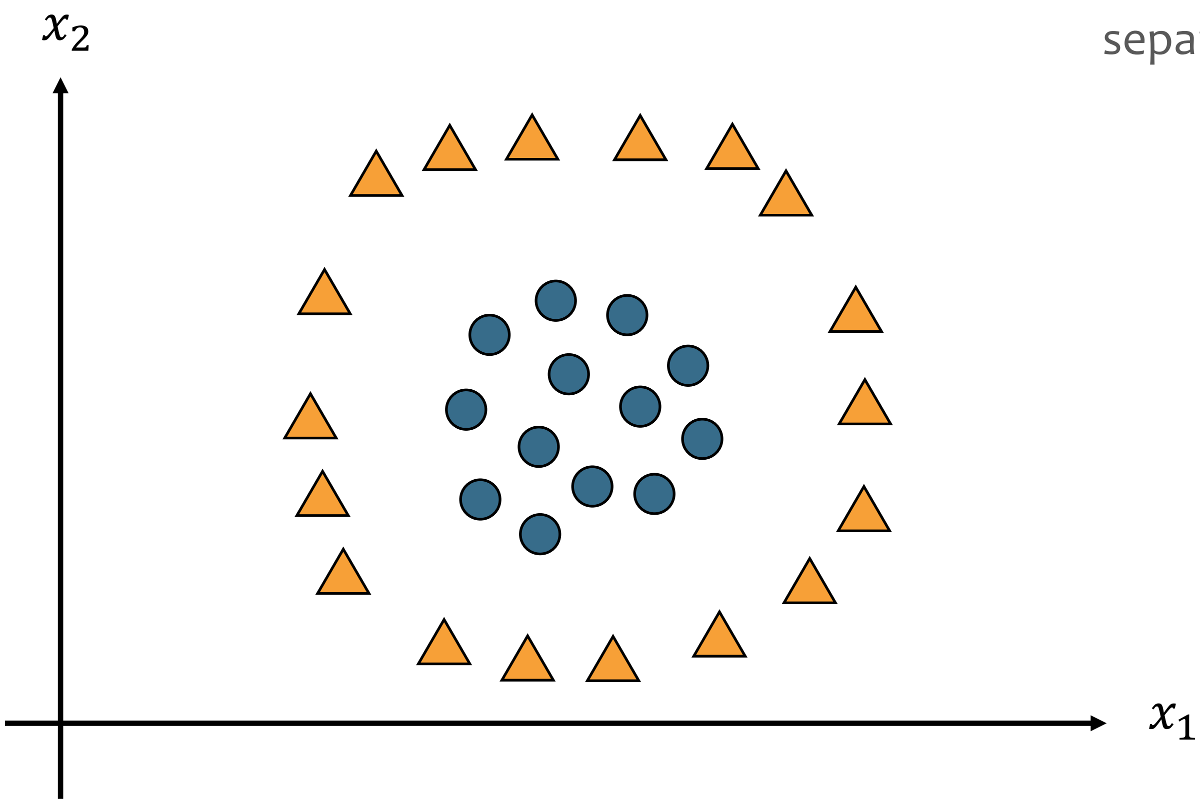
Several **classification problems** of different domains are **linearly separable**:

- **Their classes** can be separated in the feature space by **a linear model**.



Linear and Nonlinear Separability of Classes

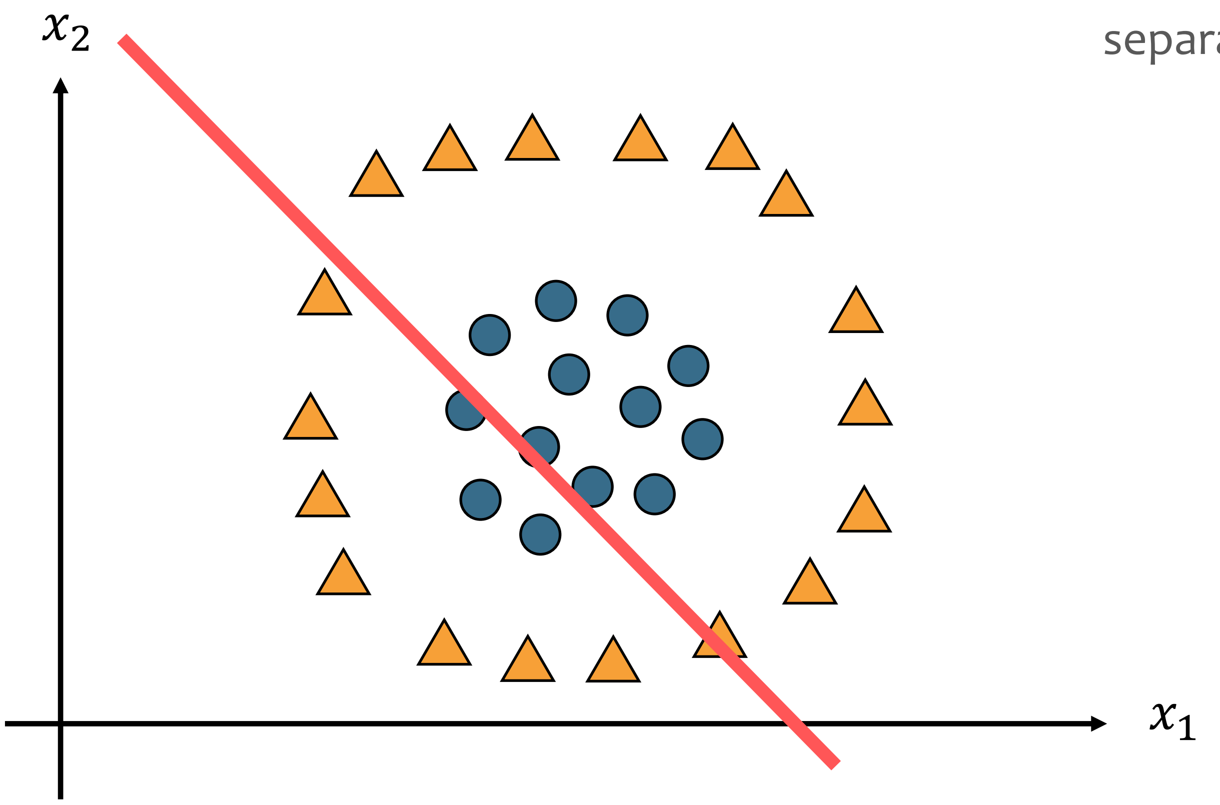
However, there are **classification problems** where **there is no** linear separability **between classes**.



There is no a linear model that separates the classes correctly.

Linear and Nonlinear Separability of Classes

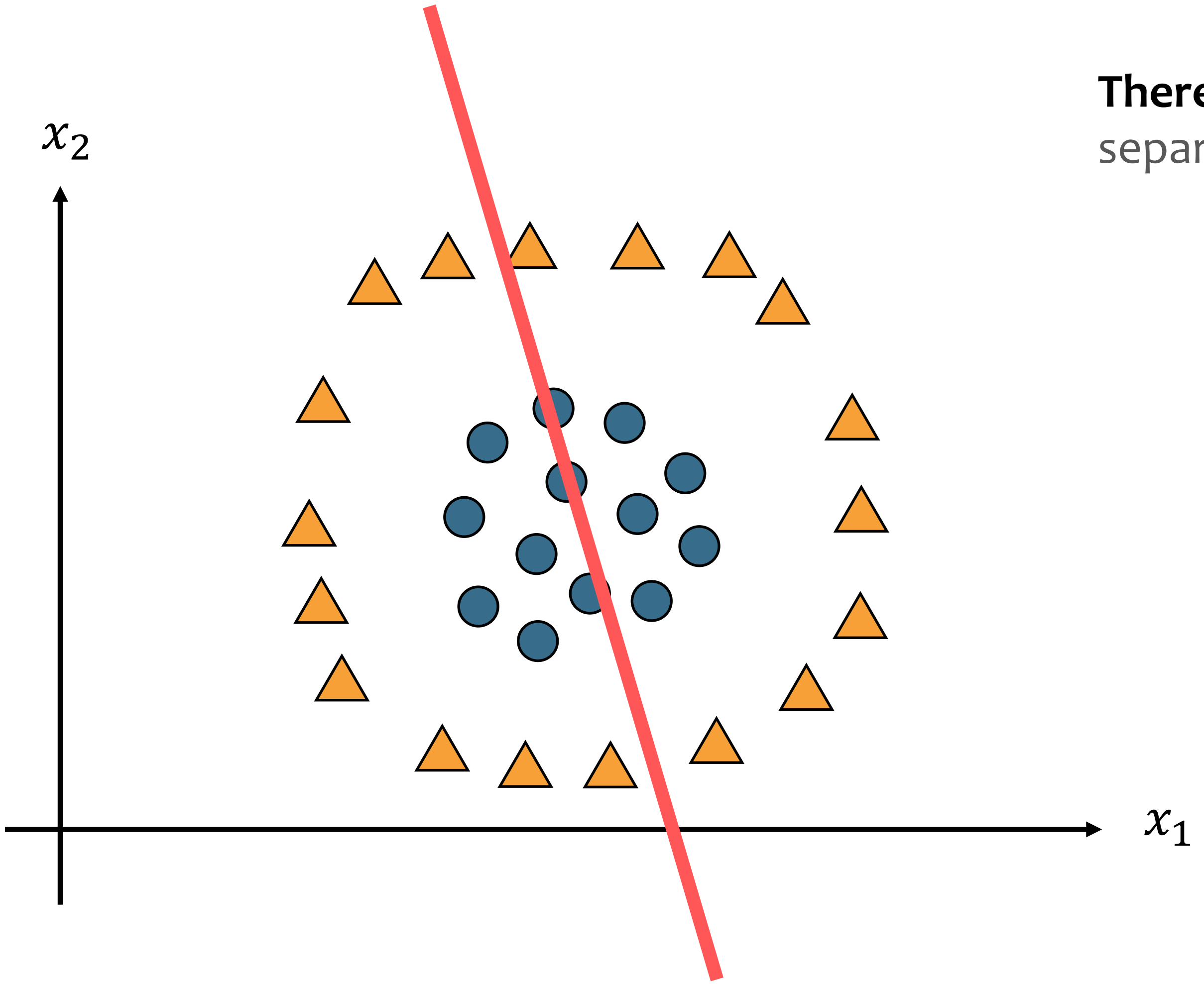
However, there are **classification problems** where **there is no** linear separability **between classes**.



There is no a linear model that separates the classes correctly.

Linear and Nonlinear Separability of Classes

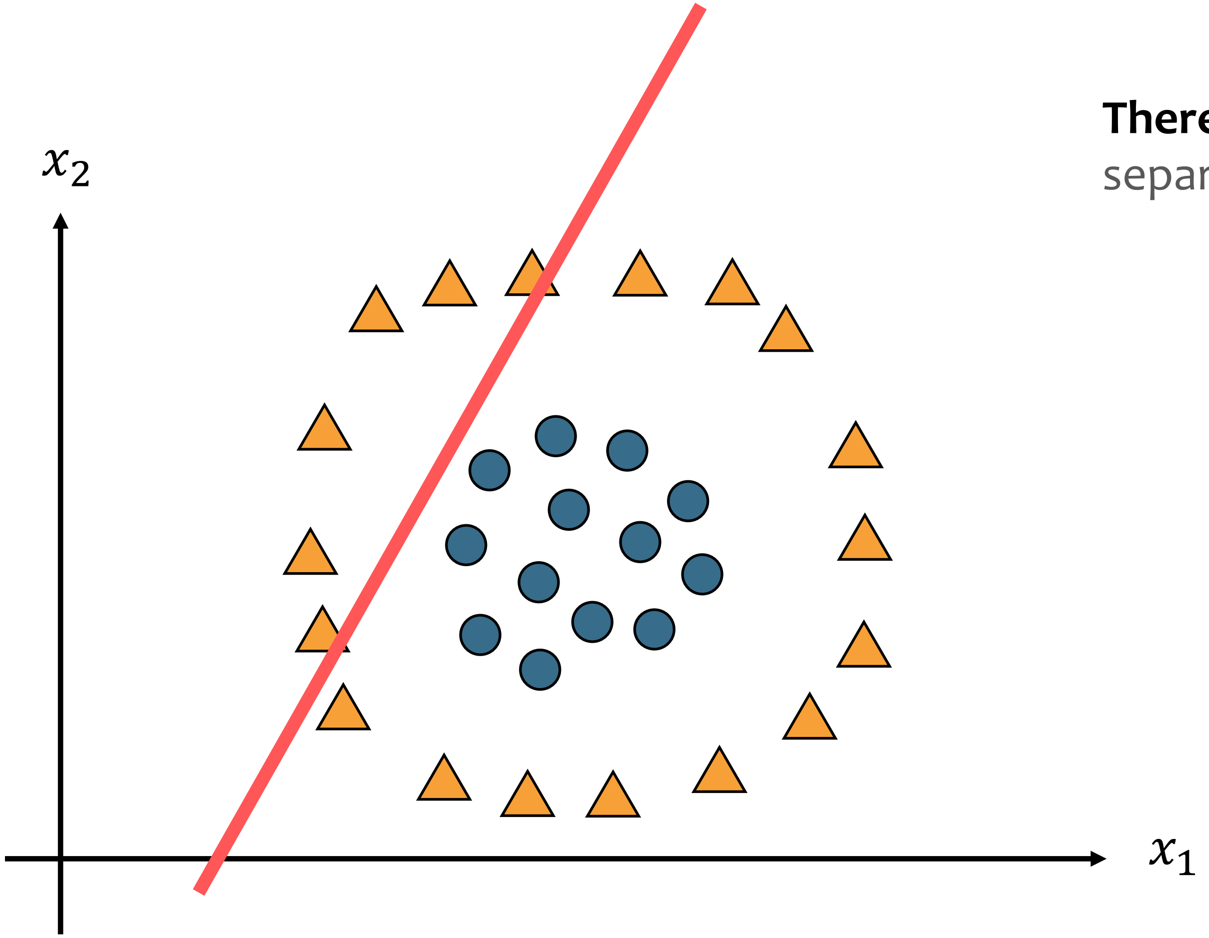
However, there are **classification problems** where **there is no linear separability between classes**.



There is no a linear model that separates the classes correctly.

Linear and Nonlinear Separability of Classes

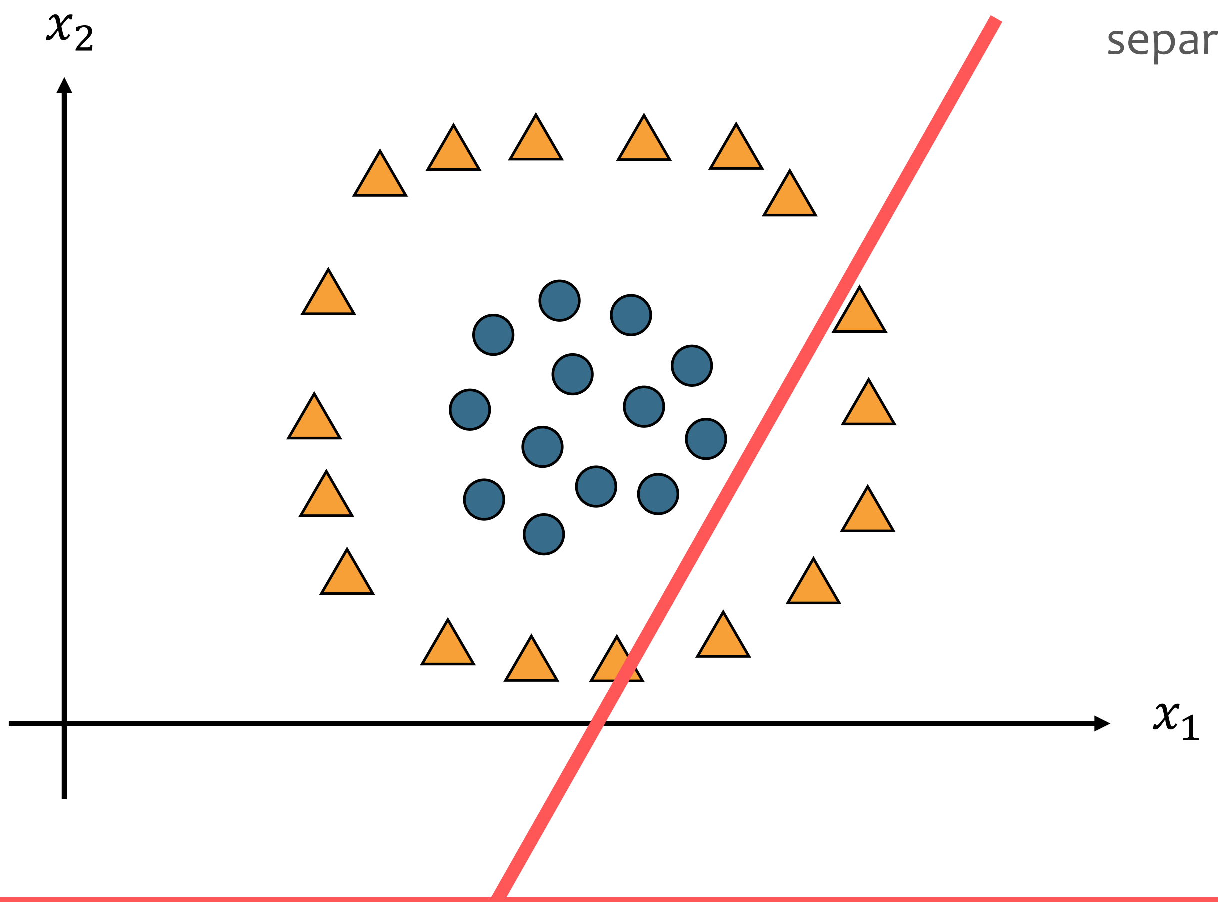
However, there are **classification problems** where **there is no linear separability between classes**.



There is no a linear model that separates the classes correctly.

Linear and Nonlinear Separability of Classes

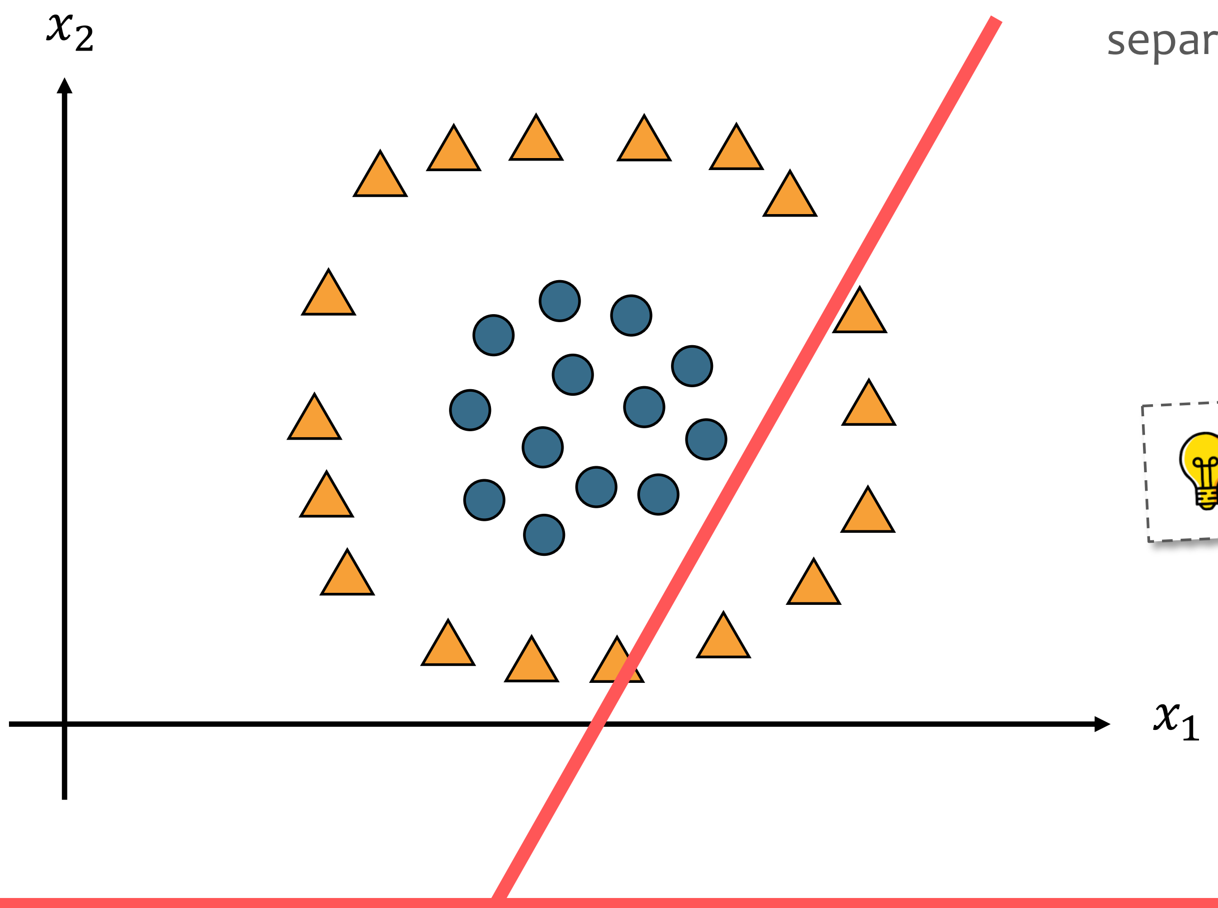
However, there are **classification problems** where **there is no linear separability between classes**.



There is no a linear model that separates the classes correctly.

Linear and Nonlinear Separability of Classes

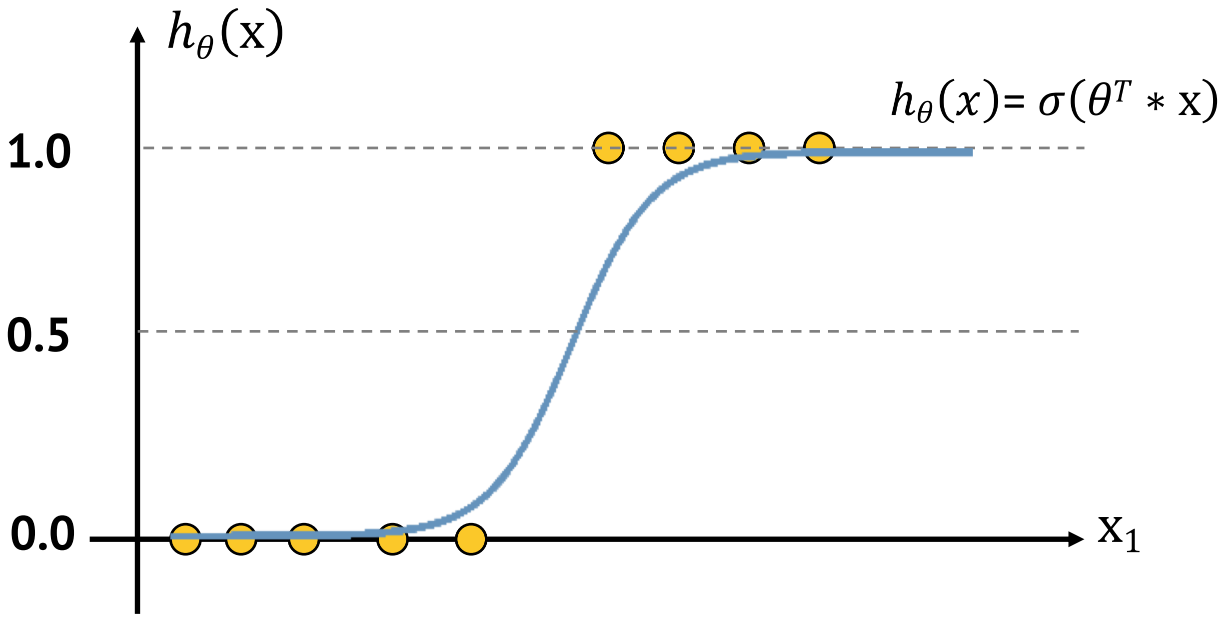
However, there are **classification problems** where **there is no linear separability between classes**.



There is no a linear model that separates the classes correctly.

These problems require **nonlinear models.**

Logistic Regression



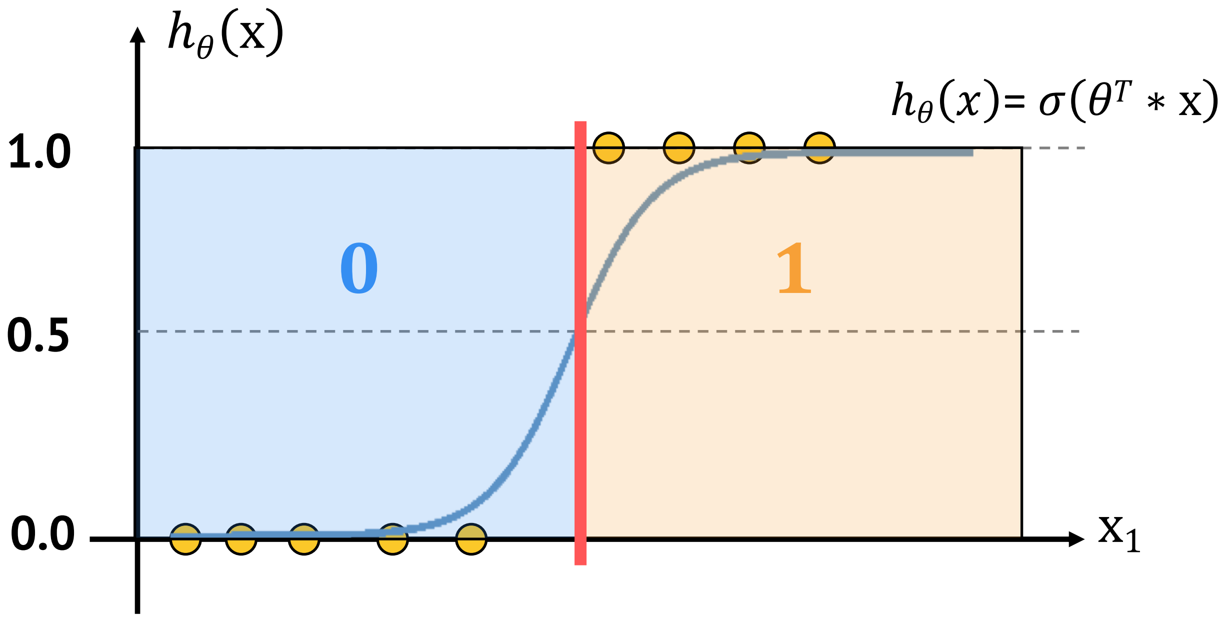
$$\boxed{h_\theta(\mathrm{x}) = \sigma(\theta^T * \mathrm{x}) = \frac{1}{1+e^{-(\theta^T * \mathrm{x})}}}$$

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \qquad \mathrm{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

$$h_\theta(\mathrm{x}) = \mathrm{P}(y = 1 | \mathrm{x}; \theta)$$

estimated probability of the observation x is **the positive class** $(y = 1)$ given a model with parameter set θ.

Logistic Regression



$$h_\theta(\mathrm{x}) = \sigma(\theta^T * \mathrm{x}) = \frac{1}{1+e^{-(\theta^T * \mathrm{x})}}$$

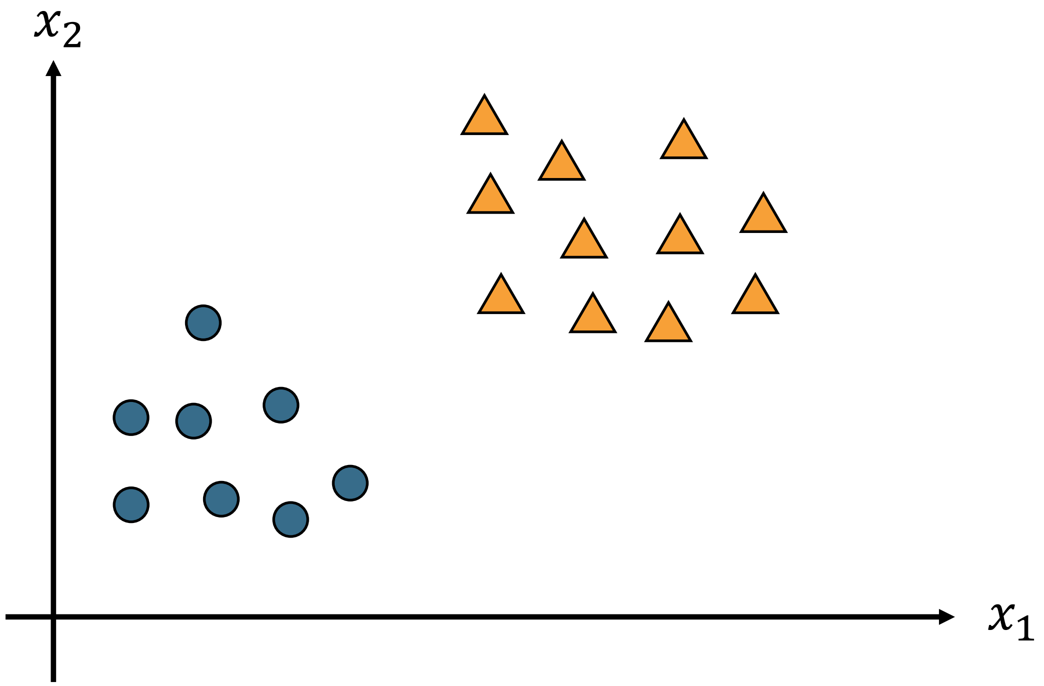
$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \quad \mathrm{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

$$h_\theta(\mathrm{x}) = \mathrm{P}(y = 1 | \mathrm{x}; \theta)$$

estimated probability of the observation x is **the positive class** $(y = 1)$ given a model with parameter set θ.

It is a **linear model** since its estimated **decision boundary** is **linear**.

Logistic Regression



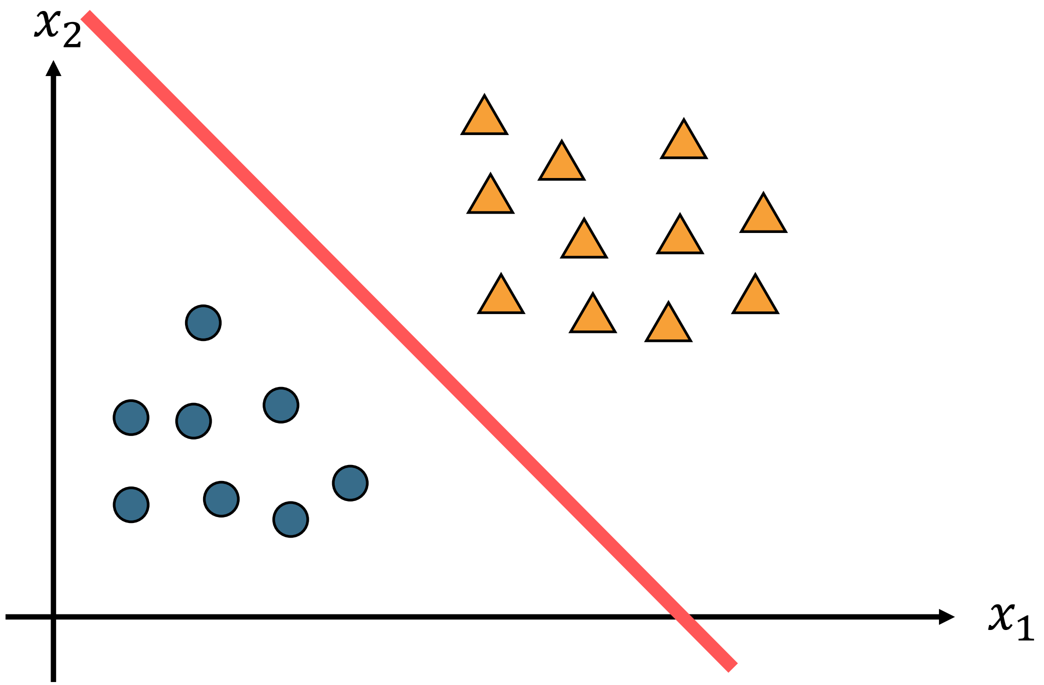
$$\boxed{h_\theta(\mathrm{x}) = \sigma(\theta^T * \mathrm{x}) = \frac{1}{1+e^{-(\theta^T * \mathrm{x})}}}$$

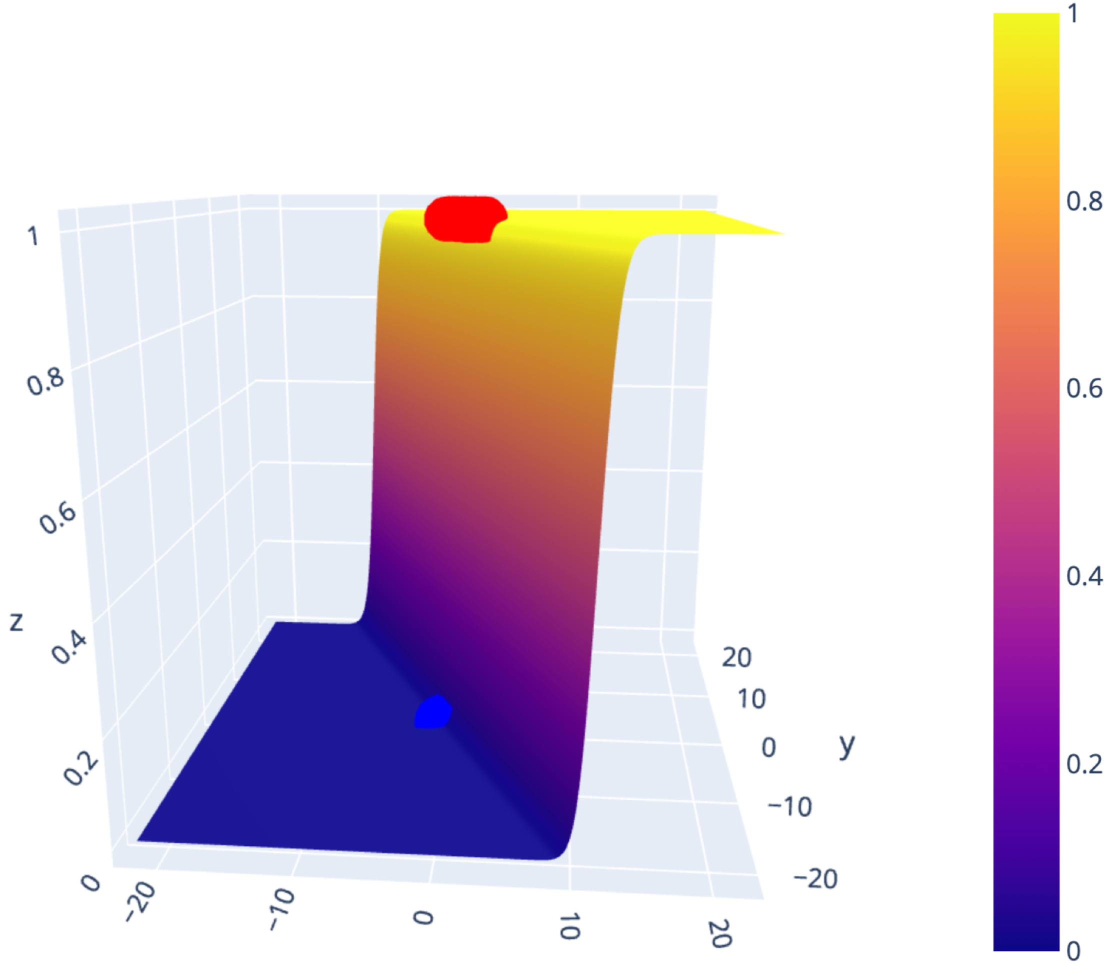
$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \qquad \mathrm{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

$$h_\theta(\mathrm{x}) = \mathrm{P}(y = 1 | \mathrm{x}; \theta)$$

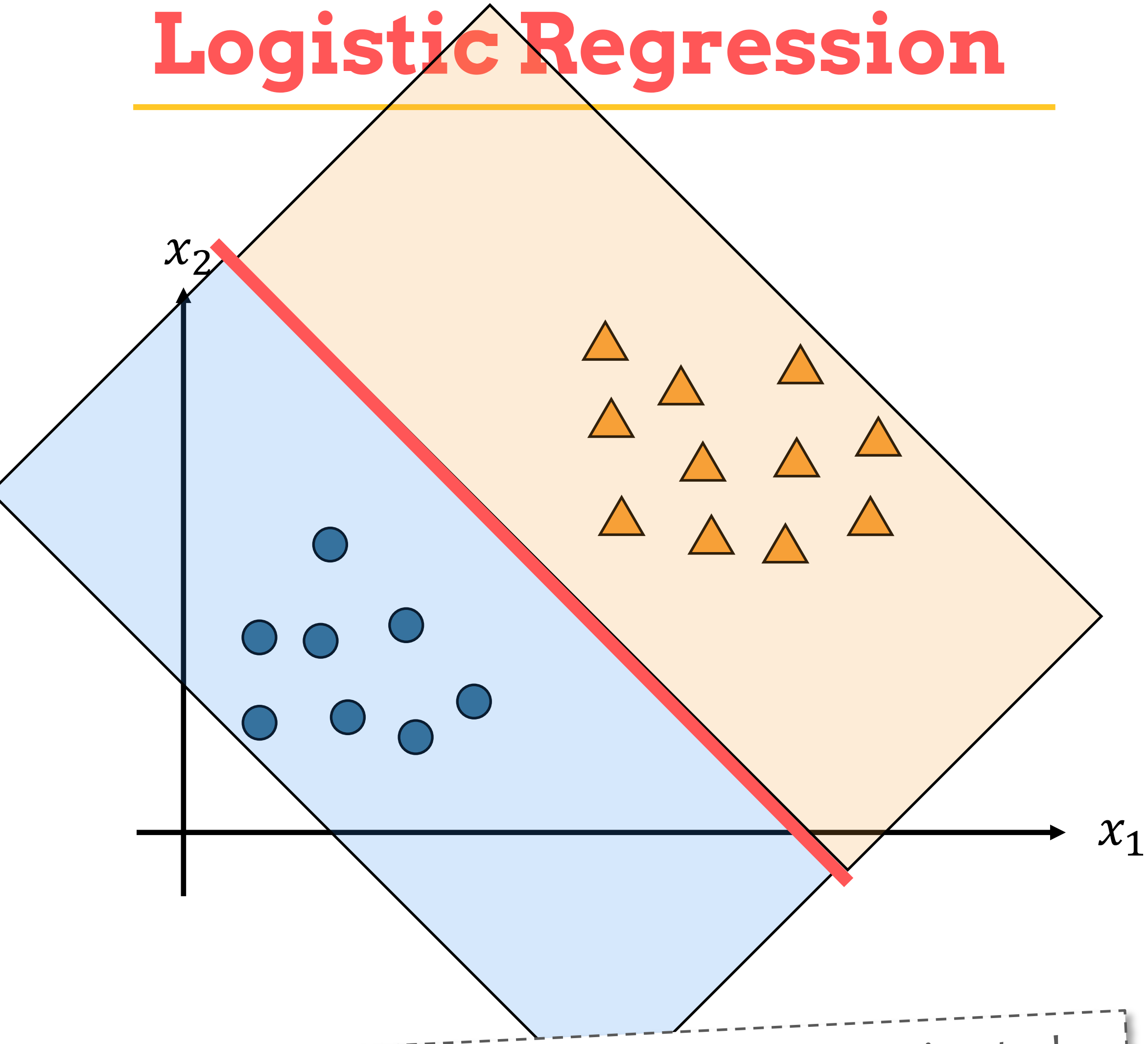
estimated probability of the observation x is **the positive class** $(y = 1)$ given a model with parameter set θ.

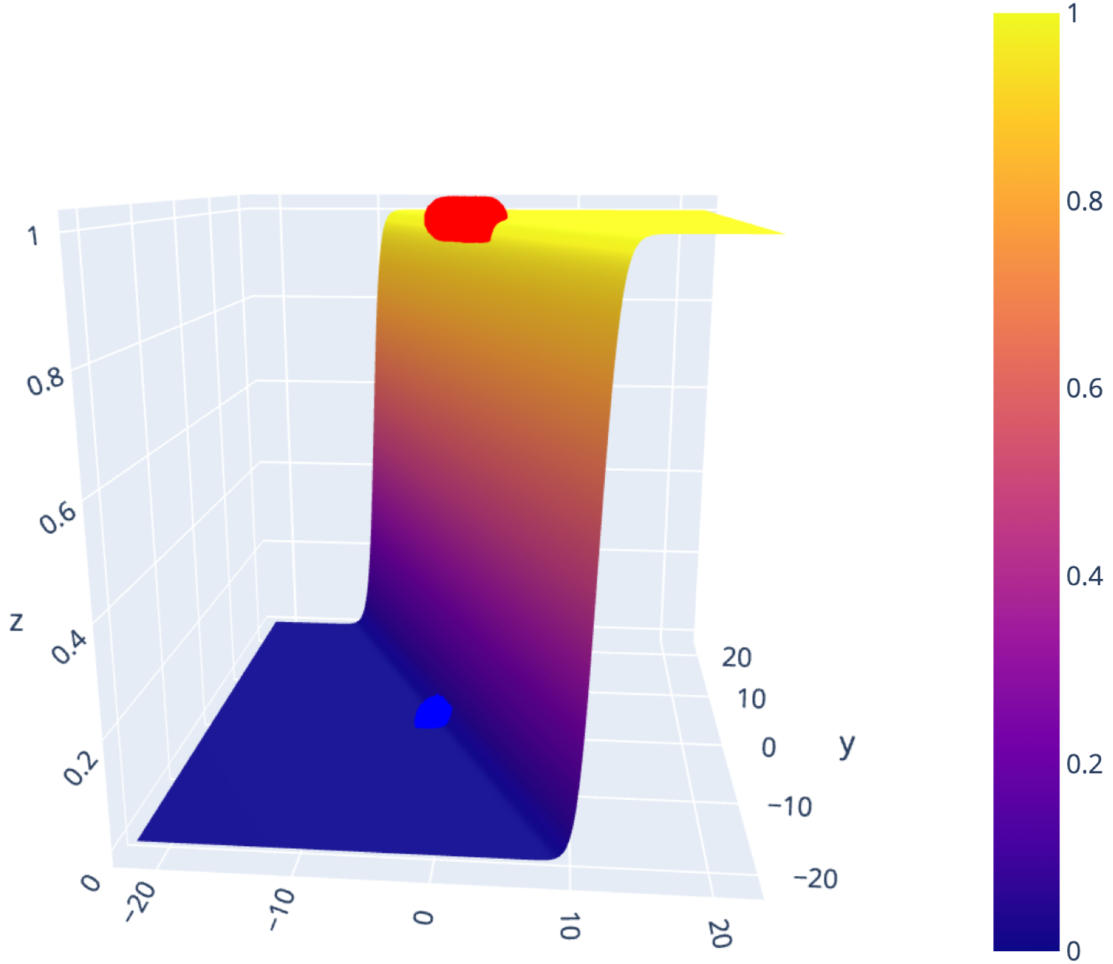
Logistic Regression





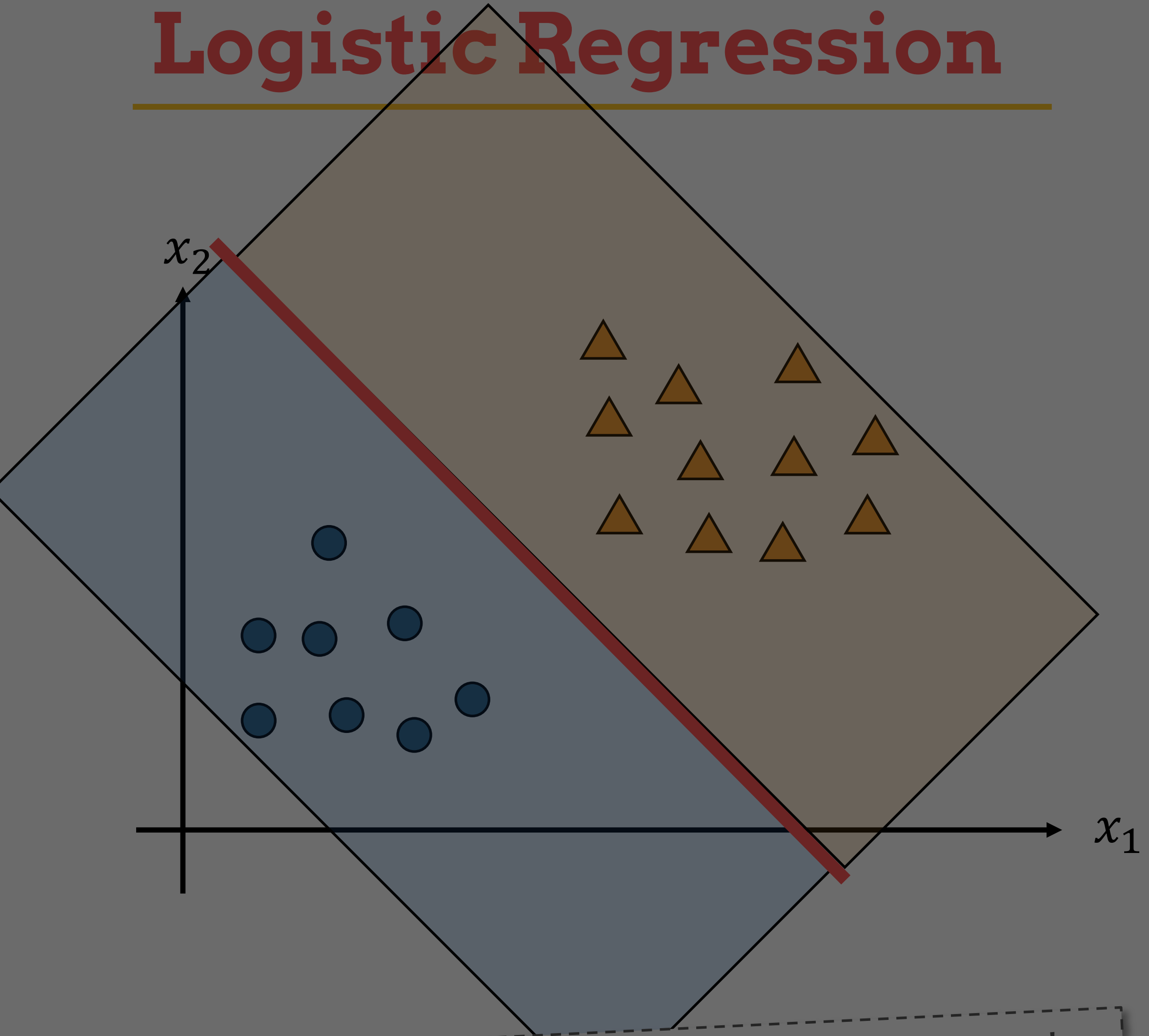
Logistic Regression



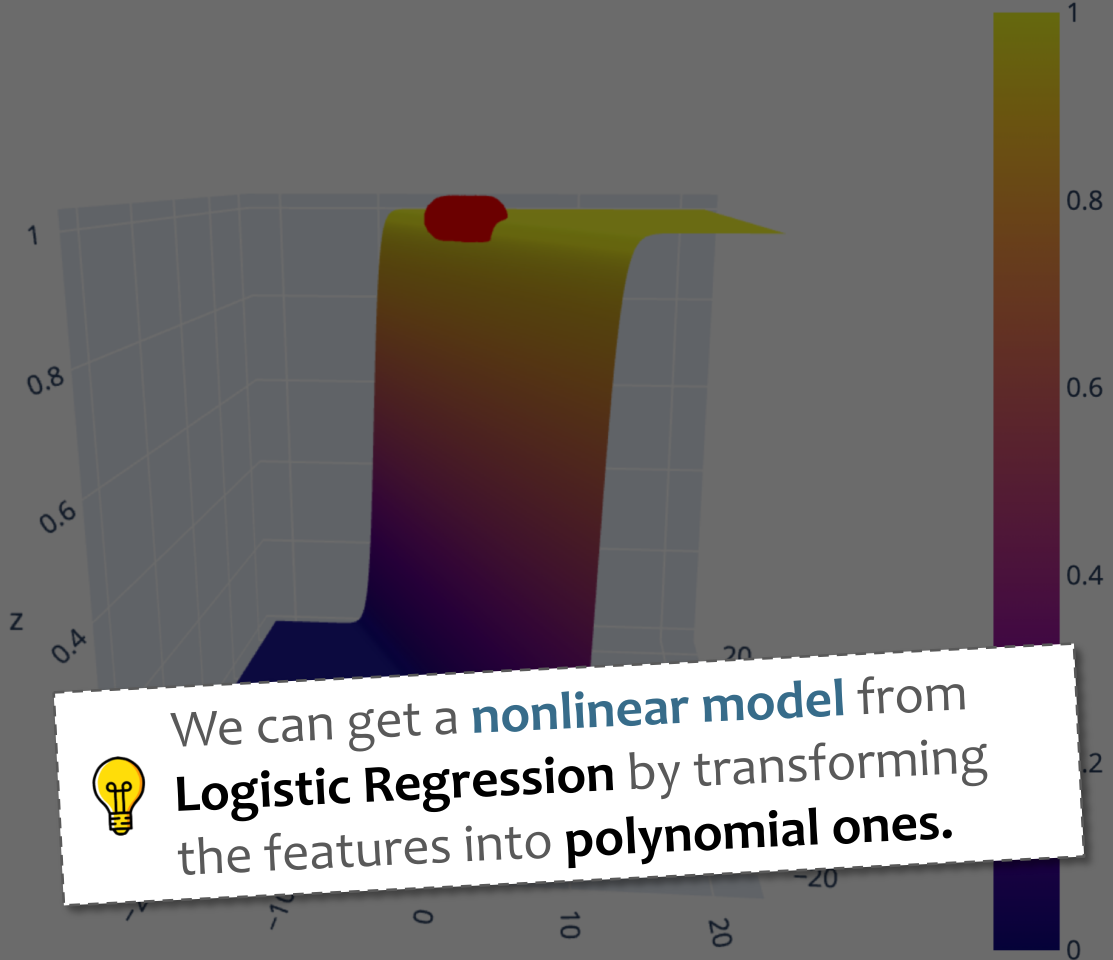


It is a **linear model** since its estimated **decision boundary** is **linear**.

Logistic Regression



💡 It is a **linear model** since its estimated **decision boundary** is **linear**.



💡 We can get a **nonlinear model** from **Logistic Regression** by transforming the features into **polynomial ones.**

Polynomial Logistic Regression

$n = 2$

two features

$$h_\theta(\mathrm{x}) = \sigma(\theta^T * \mathrm{x}) = \sigma(\theta_0 + \theta_1 \mathrm{x}_1 + \theta_2 \mathrm{x}_2) = \frac{1}{1+e^{-(\theta_0+\theta_1\mathrm{x}_1+\theta_2\mathrm{x}_2)}}$$

Polynomial Logistic Regression

$n = 2$

two features

$$h_\theta(\mathrm{x}) = \sigma(\theta^T * \mathrm{x}) = \sigma(\theta_0 + \theta_1 \mathrm{x}_1 + \theta_2 \mathrm{x}_2) = \frac{1}{1+e^{-(\theta_0+\theta_1\mathrm{x}_1+\theta_2\mathrm{x}_2)}}$$

`sklearn.preprocessing.PolynomialFeatures`

https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PolynomialFeatures.html

Polynomial Logistic Regression

$n = 2$

two features

$$h_\theta(\mathrm{x}) = \sigma(\theta^T * \mathrm{x}) = \sigma(\theta_0 + \theta_1 \mathrm{x}_1 + \theta_2 \mathrm{x}_2) = \frac{1}{1+e^{-(\theta_0+\theta_1 \mathrm{x}_1+\theta_2 \mathrm{x}_2)}}$$

`sklearn.preprocessing`.PolynomialFeatures

https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PolynomialFeatures.html

Degrees (d)**: 2**

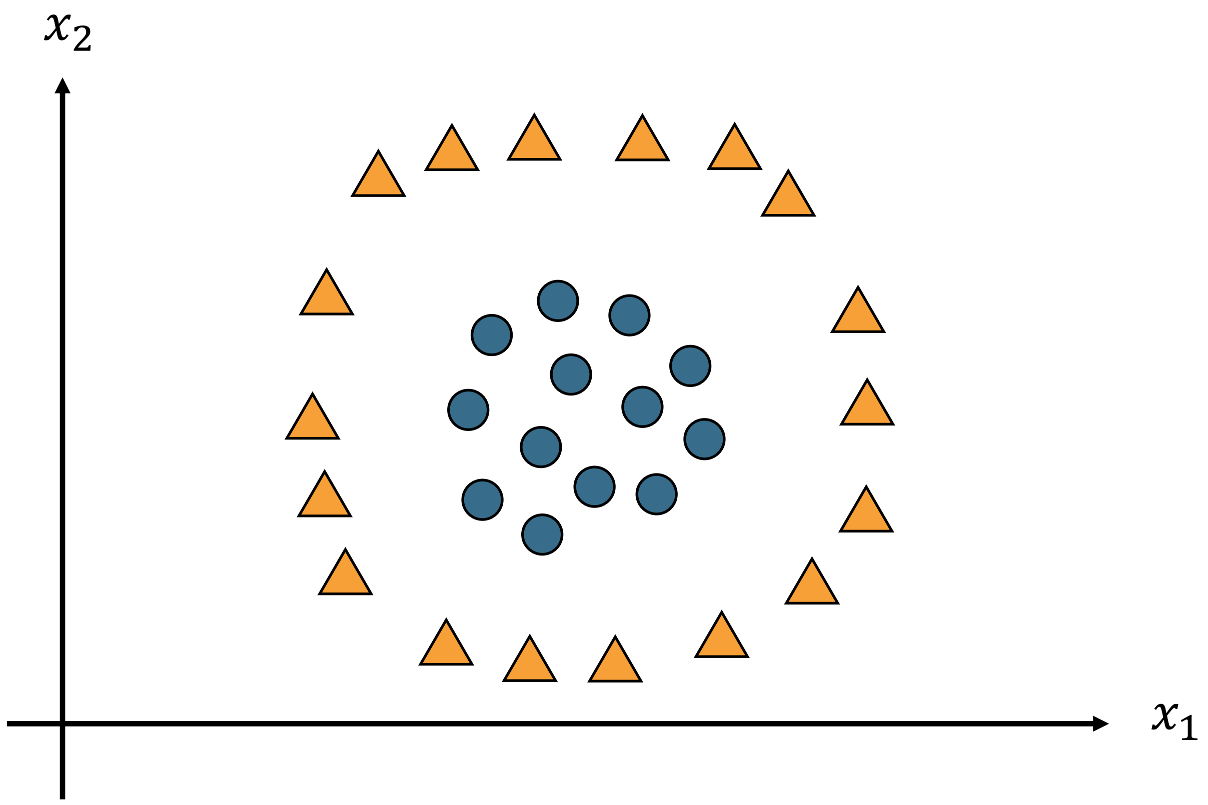
$n = 5$

$$h_\theta(\mathrm{x}) = \sigma(\theta^T * \mathrm{x}) = \sigma(\theta_0 + \theta_1 \mathrm{x}_1 + \theta_2 \mathrm{x}_2 + \theta_3 \mathrm{x}_1^2 + \theta_4 \mathrm{x}_1 \mathrm{x}_2 + \theta_5 \mathrm{x}_2^2) = \frac{1}{1+e^{-(\theta_0+\theta_1 \mathrm{x}_1+\theta_2 \mathrm{x}_2+\theta_3 \mathrm{x}_1^2+\theta_4 \mathrm{x}_1 \mathrm{x}_2+\theta_5 \mathrm{x}_2^2)}}$$

Very similar to **Polynomial Regression.**

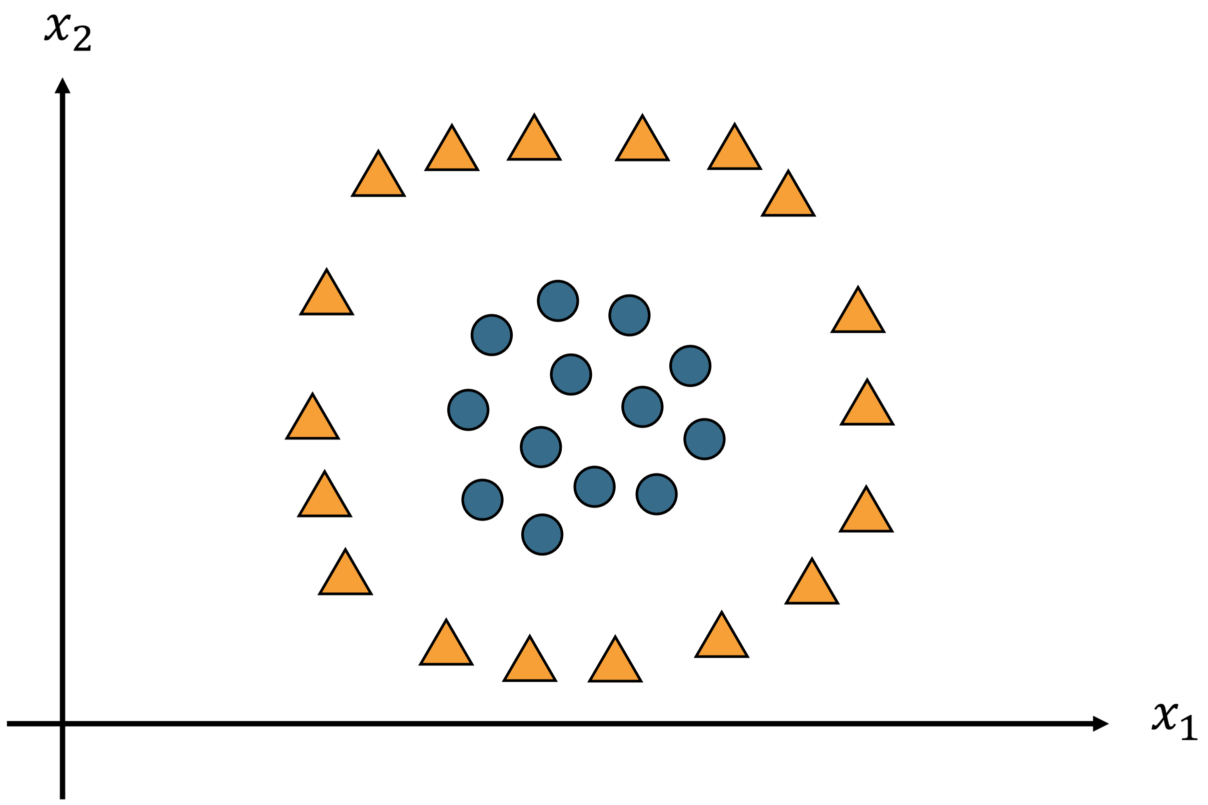
Degrees (d): 2

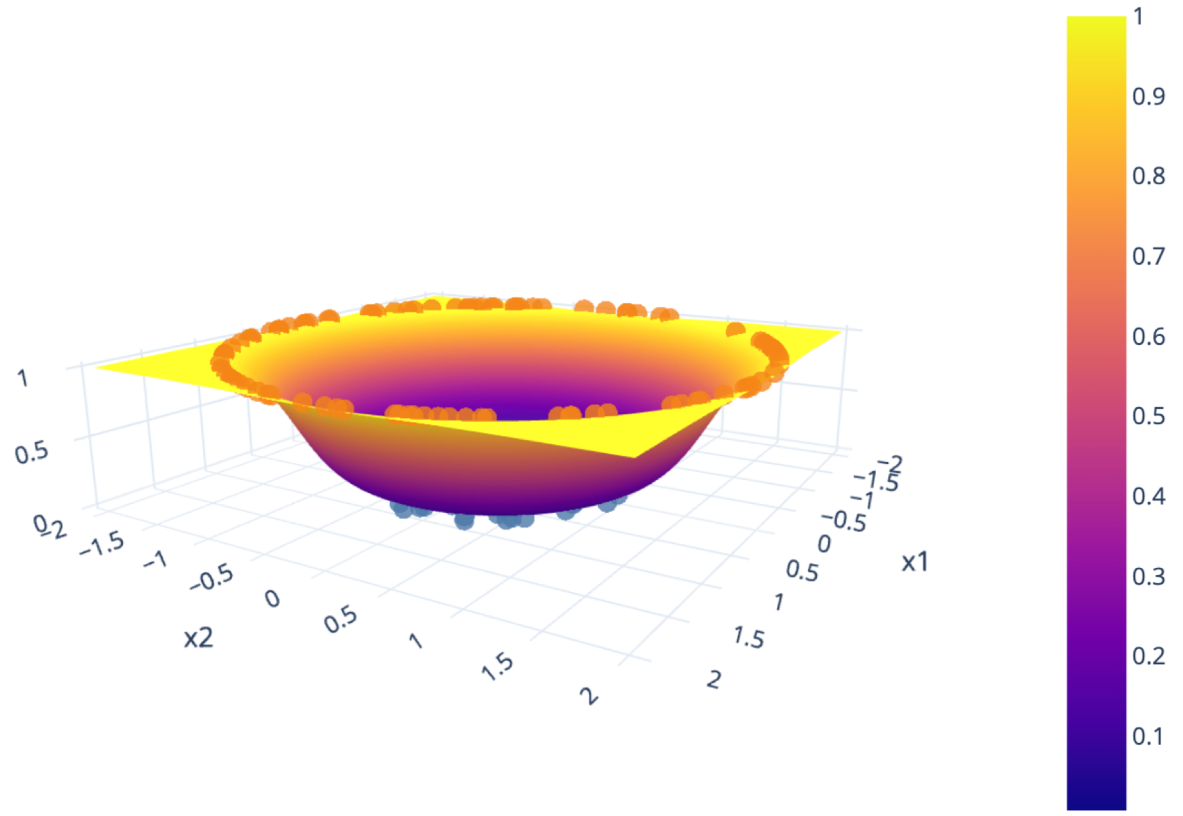
$$h_\theta(\mathrm{x}) = \sigma(\theta^T * \mathrm{x}) = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1 x_2 + \theta_5 x_2^2) = \frac{1}{1+e^{-(\theta_0+\theta_1 x_1+\theta_2 x_2+\theta_3 x_1^2+\theta_4 x_1 x_2+\theta_5 x_2^2)}}$$



Degrees (d)**: 2**

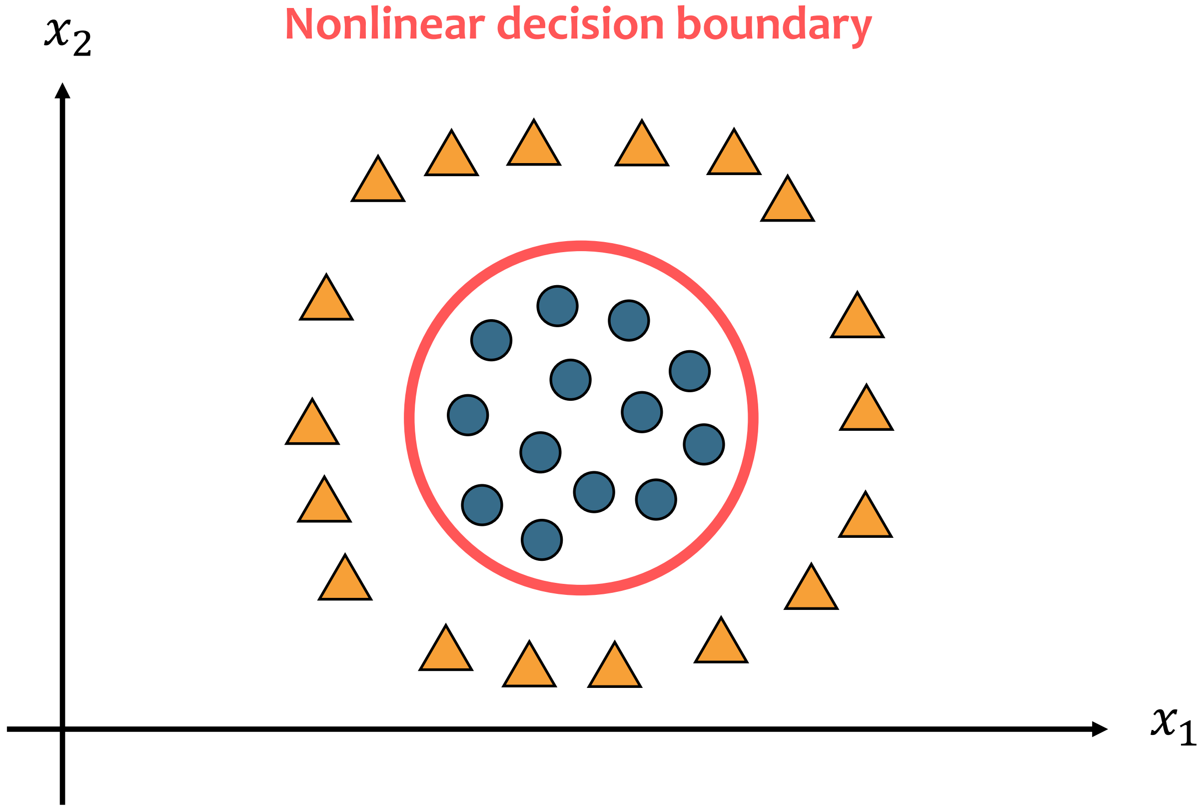
$$h_\theta(x) = \sigma(\theta^T * x) = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1 x_2 + \theta_5 x_2^2) = \frac{1}{1+e^{-(\theta_0+\theta_1 x_1+\theta_2 x_2+\theta_3 x_1^2+\theta_4 x_1 x_2+\theta_5 x_2^2)}}$$

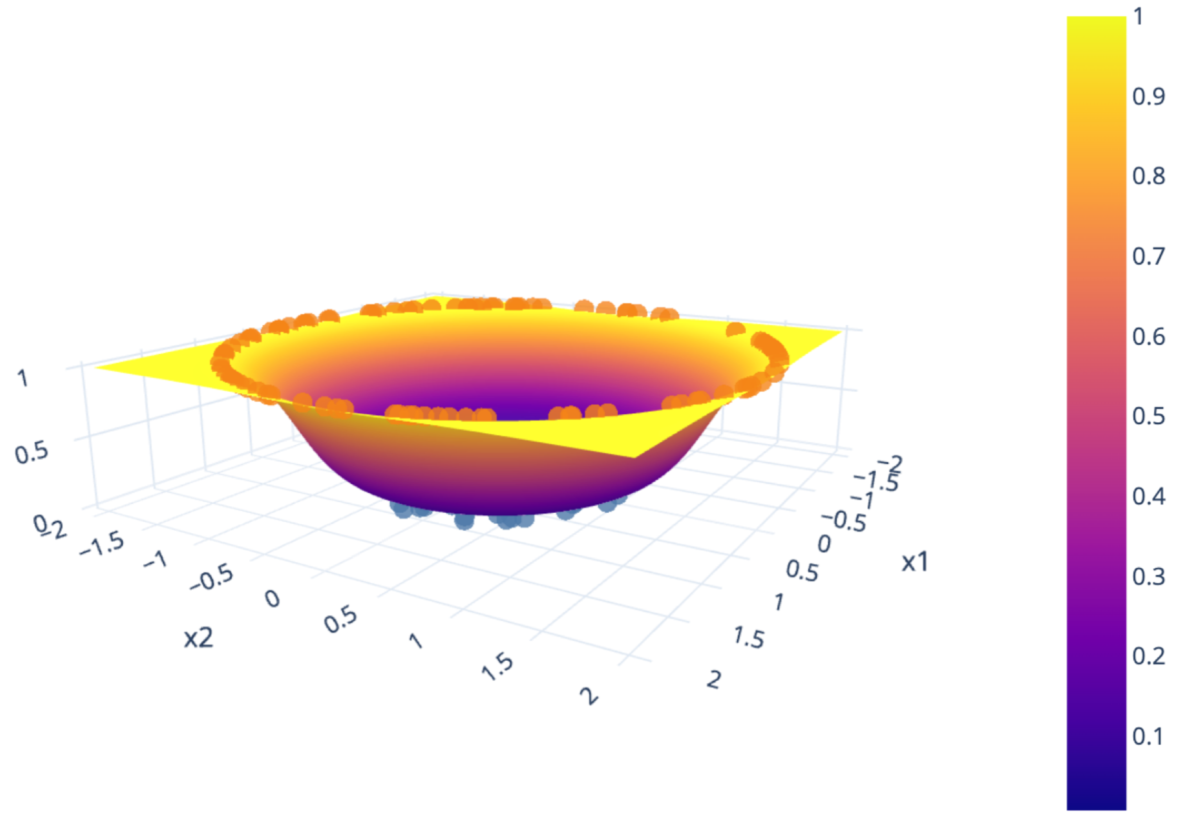




Degrees (d)**: 2**

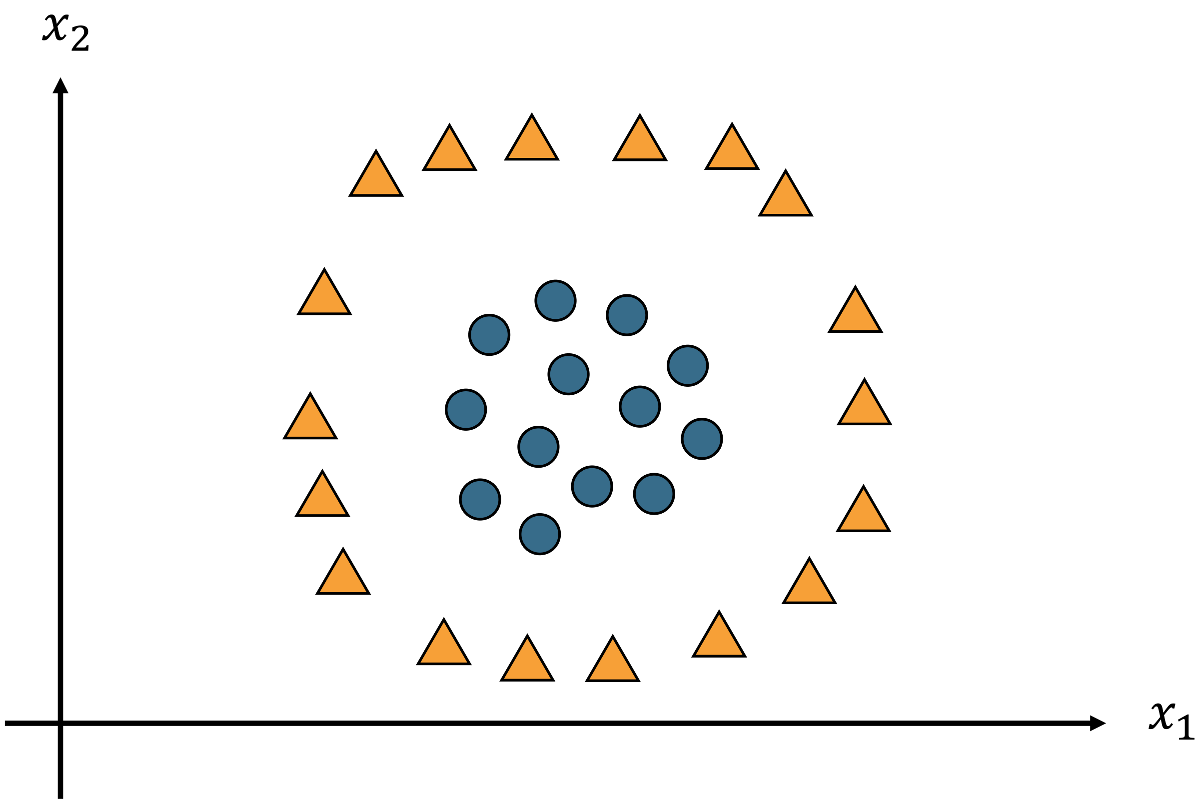
$$h_\theta(\mathrm{x}) = \sigma(\theta^T * \mathrm{x}) = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1 x_2 + \theta_5 x_2^2) = \frac{1}{1+e^{-(\theta_0+\theta_1 x_1+\theta_2 x_2+\theta_3 x_1^2+\theta_4 x_1 x_2+\theta_5 x_2^2)}}$$





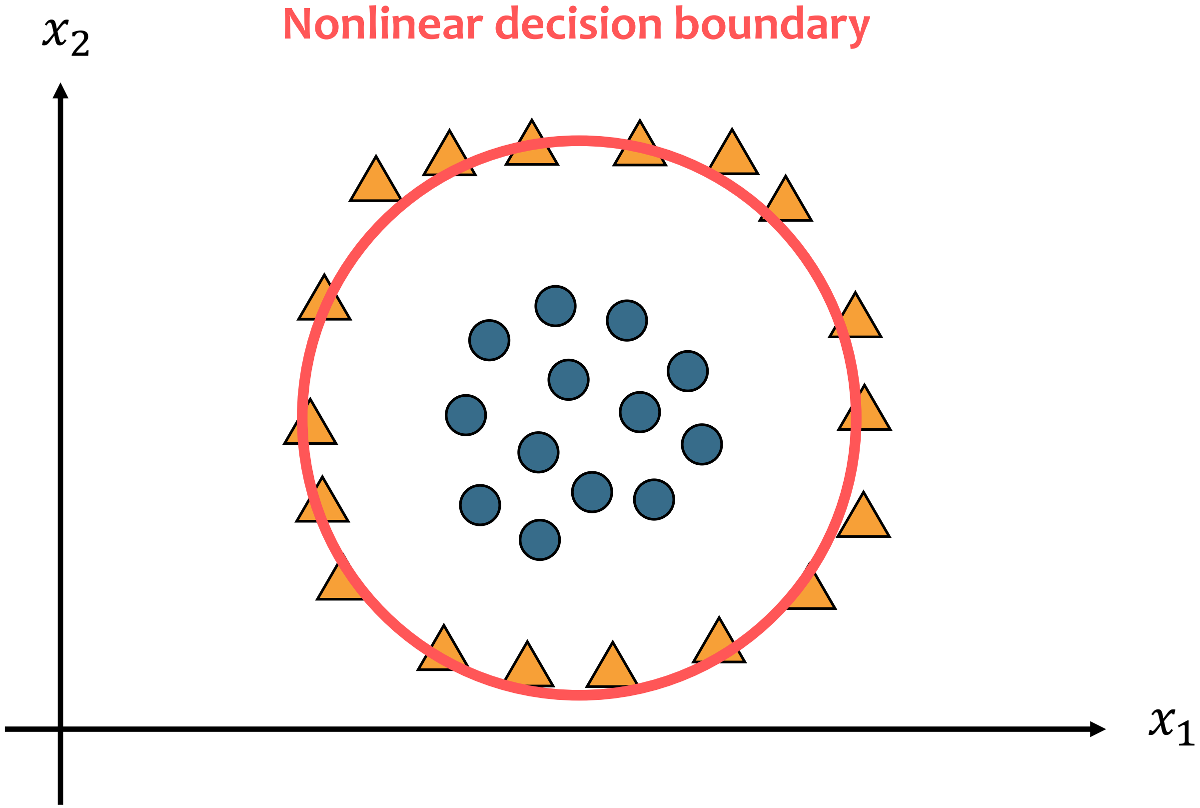
Degrees (d)**: 50**

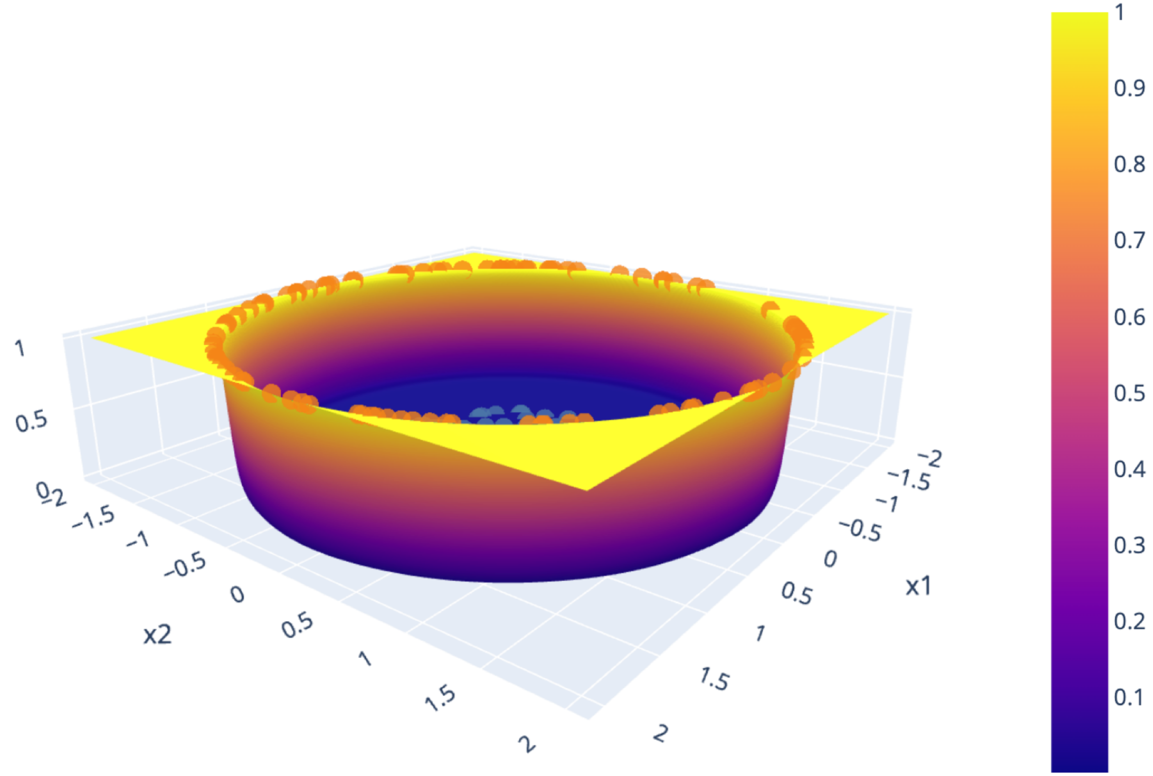
$$h_\theta(\mathrm{x}) = \sigma(\theta^T * \mathrm{x}) = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1 x_2 + \theta_5 x_2^2 + \cdots)$$



Degrees (d)**: 30**

$$h_\theta(\mathrm{x}) = \sigma(\theta^T * \mathrm{x}) = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1 x_2 + \theta_5 x_2^2 + \cdots)$$

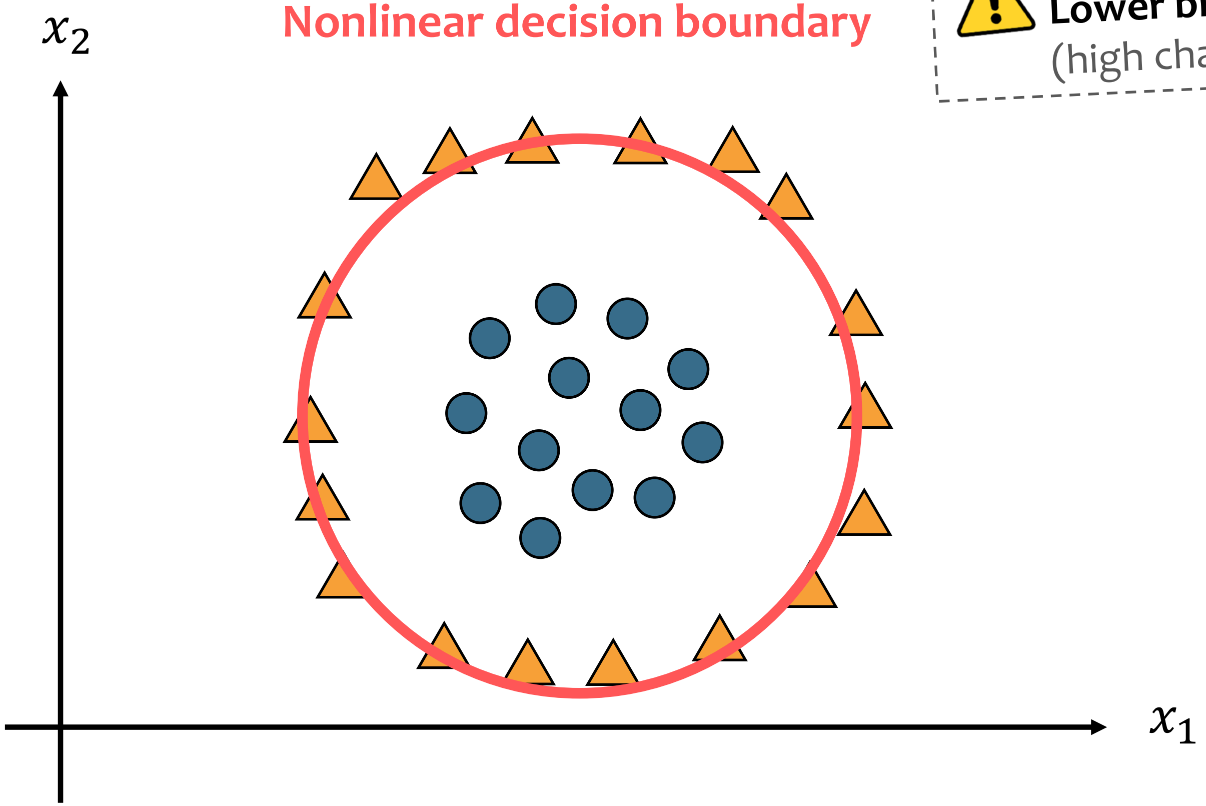


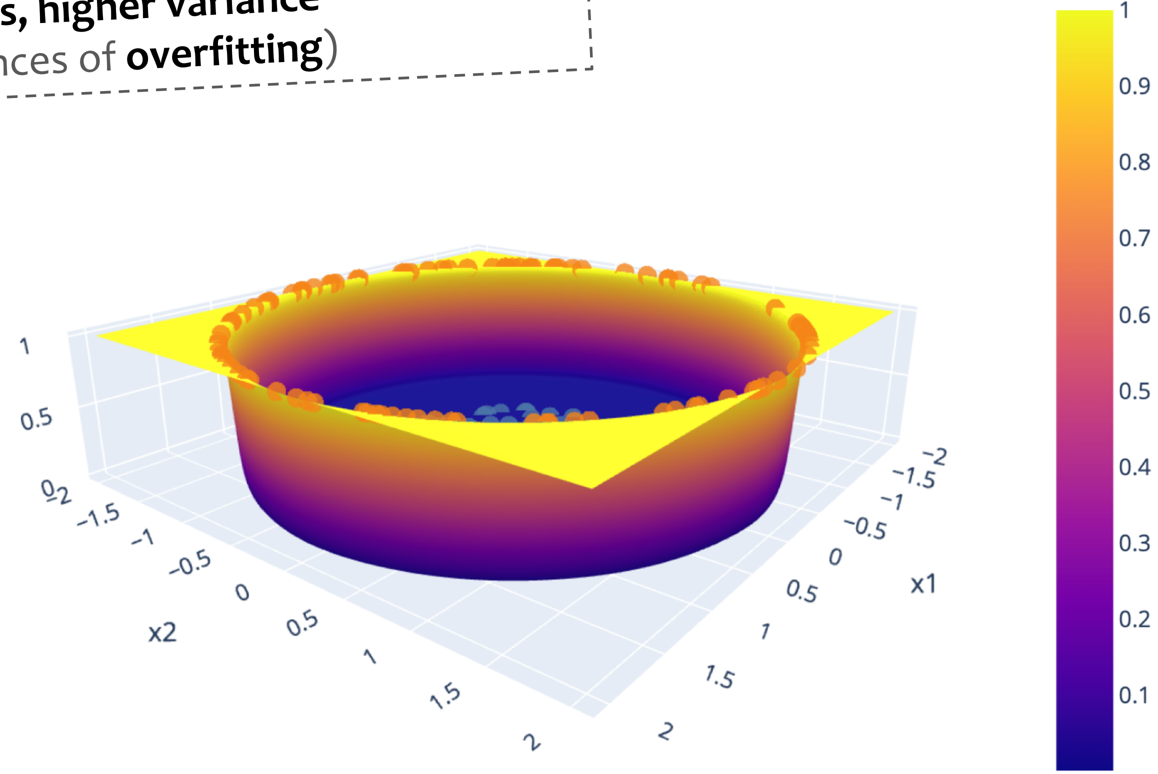


Degrees (d)**: 30**

$$h_\theta(x) = \sigma(\theta^T * x) = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1 x_2 + \theta_5 x_2^2 + \cdots)$$

A really **more complex nonlinear model**:
Lower bias, higher variance
(high chances of **overfitting**)





Let's code it!

Polynomial Logistic Regression

Degree = 2

```
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import PolynomialFeatures
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression

pol_log_reg_clf = Pipeline([
    ('pol_feats', PolynomialFeatures(degree=2, include_bias=False)),
    ("std_scaler", StandardScaler()),
    ("log_reg", LogisticRegression(random_state=42)),
])

pol_log_reg_clf.fit(X_train, y_train)
```

Aprendizado de Máquina e Reconhecimento de Padrões

Polynomial Logistic Regression

Prof. Dr. Samuel Martins (Samuka)
samuel.martins@ifsp.edu.br

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
São Paulo
Campus Campinas