

Machine Learning

Problemas de Clasificación

Jose Luis Paniagua Jaramillo
jlpaniagua@uao.edu.co

1 Problemas de Clasificación

- Clasificación
- Perceptron
- Logistic Regression
- SVM
- Kernel SVM

2 Medidas de Desempeño

- Matriz de Confusión

3 Referencias

1 Problemas de Clasificación

- Clasificación
- Perceptron
- Logistic Regression
- SVM
- Kernel SVM

2 Medidas de Desempeño

- Matriz de Confusión

3 Referencias

1 Problemas de Clasificación

- Clasificación
- Perceptron
- Logistic Regression
- SVM
- Kernel SVM

2 Medidas de Desempeño

- Matriz de Confusión

3 Referencias

- La clasificación es un tipo de aprendizaje supervisado, donde el objetivo es predecir resultados con valores discretos (**variables categóricas o etiquetas**).
- Con base en datos pasados, se busca predecir la categoría o clase a la cual pertenece un dato nunca antes visto.

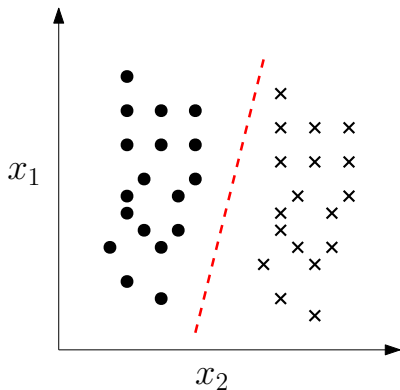
Ejemplo

filtro de correos spam.

Podemos entrenar un modelo con un lote de correos electrónicos etiquetados debidamente como spam y no spam para predecir si un nuevo correo que llega pertenece a una categoría o la otra.

Clasificación

Clasificación Binaria



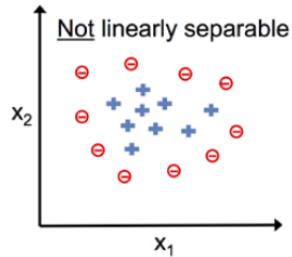
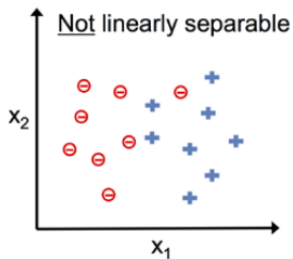
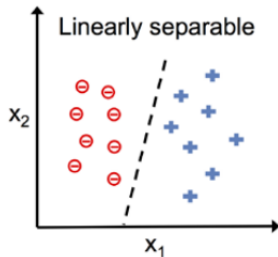
Datos:

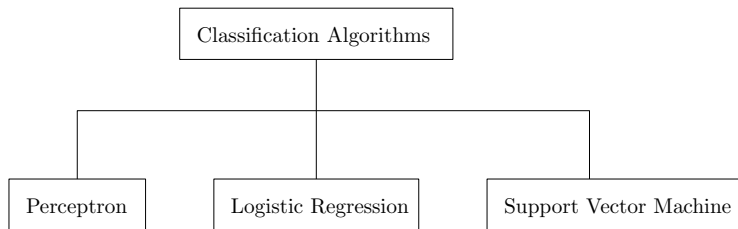
- clase 1: círculos.
- clase 2: cruces.
- 18 datos(training samples) para cada una de las clases.

- Cuando la tarea de clasificación requiere clasificar 2 clases, el problema se denomina **Clasificación Binaria**.
- Cuando la tarea de clasificación requiere clasificar 3 o mas clases, el problema se denomina **Clasificación Multiclase**.
- Cuando los datos se pueden separar por una recta, el problema se denomina ademas **Linealmente Separable**

Clasificación

Datos No Separables Linealmente





- Estos algoritmos permiten realizar clasificación binaria.
- Estos algoritmos naturalmente están diseñados para clasificar datos que se pueden separar linealmente. Sin embargo, se pueden modificar para clasificar datos separables de forma no lineal.

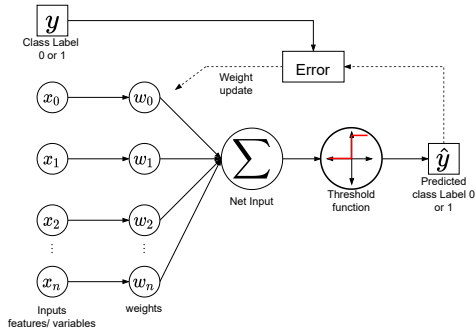
1 Problemas de Clasificación

- Clasificación
- **Perceptron**
- Logistic Regression
- SVM
- Kernel SVM

2 Medidas de Desempeño

- Matriz de Confusión

3 Referencias



$$X = [x_0, x_1, \dots, x_n]$$

$$W = [w_0, w_1, \dots, w_n]$$

$$f(w, x) = \sum_{i=1}^n w_i x_i = W^T X$$

$$output = \begin{cases} 1, & f(w, x) \geq 0 \\ 0, & f(w, x) < 0 \end{cases}$$

Algoritmo de Aprendizaje

El entrenamiento del perceptron se realiza mediante la función de costo MSE y el Gradiente Descendente.

Perceptron II

Ejemplo:

Haberman's Survival Data Set

<https://archive.ics.uci.edu/ml/datasets/Haberman%27s+Survival>



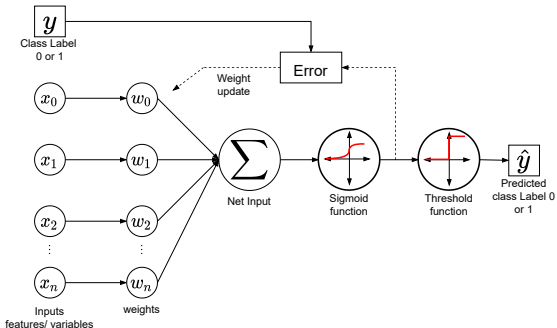
1 Problemas de Clasificación

- Clasificación
- Perceptron
- **Logistic Regression**
- SVM
- Kernel SVM

2 Medidas de Desempeño

- Matriz de Confusión

3 Referencias



$$X = [x_0, x_1, \dots, x_n]$$

$$W = [w_0, w_1, \dots, w_n]$$

$$SUM = \sum_{i=1}^n w_i x_i = W^T X$$

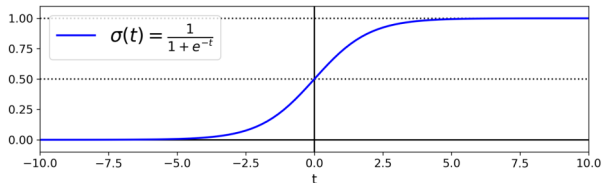
Sigmoid:

$$output = \frac{1}{1 + e^{w^T x}}$$

La regresión logisitca se usa para estimar la probabilidad de que un dato pertenezca a una clase en particular.

Sigmoid Function

$$\hat{p} = h_W(X) = \frac{1}{1 + e^{-t}} = \sigma(t)$$



La Regresión Logística toma como entrada una combinación lineal de las variables y genera como salida un valor numérico entre 0 y 1.

$$\hat{y} = \begin{cases} 0 & \text{if } \hat{p} < 0,5 \\ 1 & \text{if } \hat{p} \geq 0,5 \end{cases}$$

Algoritmo de Aprendizaje

- La función de costo para la regresión logística es una función probabilística.

$$c(\mathbf{w}) \begin{cases} -\log(\hat{p}) & \text{if } y = 1 \\ -\log(1 - \hat{p}) & \text{if } y = 0 \end{cases}$$

- La función de costo es convexa, lo cual garantiza que el Gradiente Descendente puede encontrar el mínimo global.

$$J(\boldsymbol{\theta}) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(\hat{p}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{p}^{(i)})]$$

Logistic Regression IV

Ejemplo:

banknote authentication Data Set

<https://archive.ics.uci.edu/ml/datasets/banknote+authentication>



1 Problemas de Clasificación

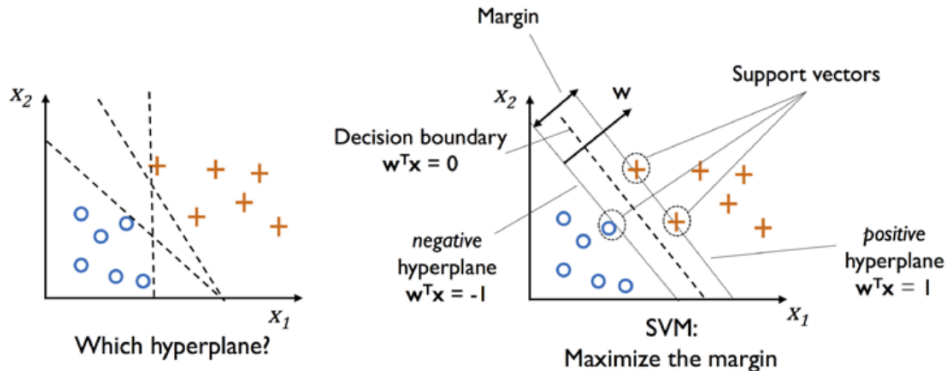
- Clasificación
- Perceptron
- Logistic Regression
- **SVM**
- Kernel SVM

2 Medidas de Desempeño

- Matriz de Confusión

3 Referencias

Support Vector Machine (SVM) I



- El objetivo es encontrar un **margen** de separación, cuyo ancho queda determinado por los datos mas cercanos de cada clase.
- La linea que maximiza el margen es la optima para el modelo.

Algoritmo de Aprendizaje

- La función de costo de una SVM surge de la distancia entre los planos que definen el margen de la frontera de decision.

$$w_0 + w^T x_{pos} = 1$$

$$w_0 + w^T x_{neg} = -1$$

$$w^T (x_{pos} - x_{neg}) = 2$$

$$\frac{w^T (x_{pos} - x_{neg})}{\|w\|} = \frac{2}{\|w\|}$$

- El objetivo es maximizar la distancia (margen) entre los planos negativo y positivo.

Support Vector Machine (SVM) III

Ejemplo:

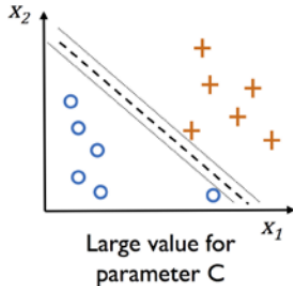
Pima Indians Diabetes Dataset

https:
`//raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.csv`

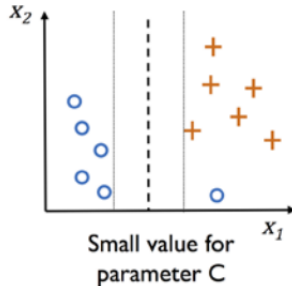


Hiperparametros

Son valores de configuraciones utilizadas durante el proceso de entrenamiento de un modelo de aprendizaje automático.



$C = 100$



$C = 1$

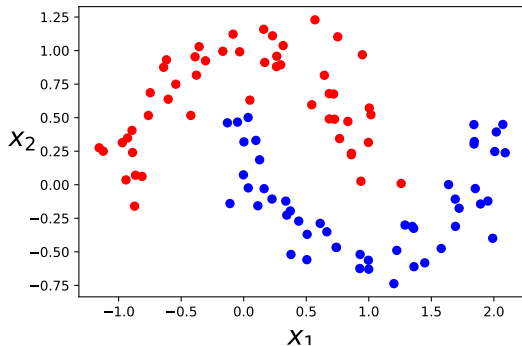
1 Problemas de Clasificación

- Clasificación
- Perceptron
- Logistic Regression
- SVM
- **Kernel SVM**

2 Medidas de Desempeño

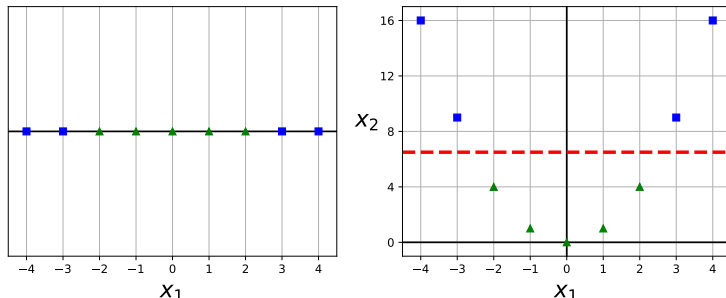
- Matriz de Confusión

3 Referencias



Un clasificador lineal solo puede separar puntos por medio de una línea recta.

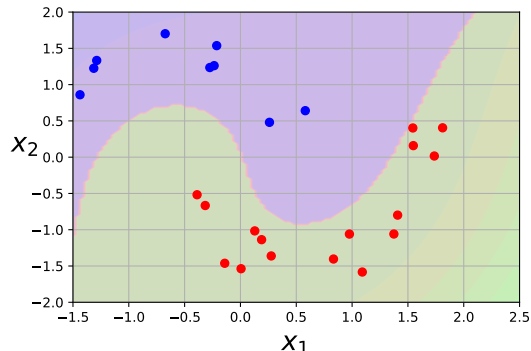
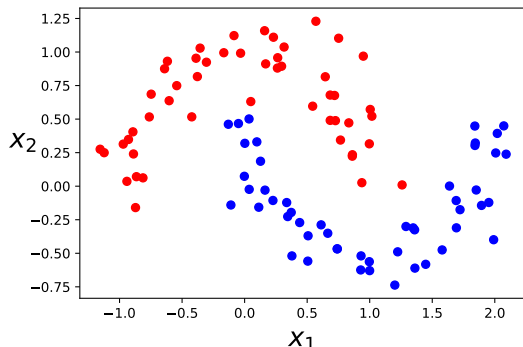
Una forma de utilizar clasificadores lineales para problemas no linealmente separables es añadir mas variables a partir de las ya existentes pero aumentando su grado.



$$X_2 = X_1^2$$

kernel trick

el *kernel trick* hace posible obtener los mismos resultados que se obtienen al añadir mas variables (polynomial features)



1 Problemas de Clasificación

- Clasificación
- Perceptron
- Logistic Regression
- SVM
- Kernel SVM

2 Medidas de Desempeño

- Matriz de Confusión

3 Referencias

1 Problemas de Clasificación

- Clasificación
- Perceptron
- Logistic Regression
- SVM
- Kernel SVM

2 Medidas de Desempeño

- Matriz de Confusión

3 Referencias

Matriz de Confusión I

Actual	Positive	Negative
	TP	FN
Negative	FP	TN
	Positive	Negative
Predicted		

- **TN: true negatives** es el numero de datos clasificados por el modelo como **negativos** y son realmente **negativos**.
- **FN: false negatives** es el numero de datos clasificados por el modelo como **negativo**, pero son realmente **positivos**.
- **TP: true positives** es el numero de datos clasificados por el modelo como **positivos** y son realmente **positivos**.
- **FP: false positives:** es el numero de datos clasificados por el modelo como **positivos** y son realmente **negativos**.

Precision

Mide la precisión de las predicciones positivas.

$$precision = \frac{TP}{TP + FP}$$

Recall (sensibilidad)

Mide la proporción de instancias positivas que el clasificador detecta correctamente

$$recall = \frac{TP}{TP + FN}$$

- 1 La Matriz de Confusión
- 2 La Precisión
- 3 La Sensibilidad

1 Problemas de Clasificación

- Clasificación
- Perceptron
- Logistic Regression
- SVM
- Kernel SVM

2 Medidas de Desempeño

- Matriz de Confusión

3 Referencias



Aurélien Géron.

Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.

O'Reilly Media, 2019.

<https://scikit-learn.org/stable/index.html>