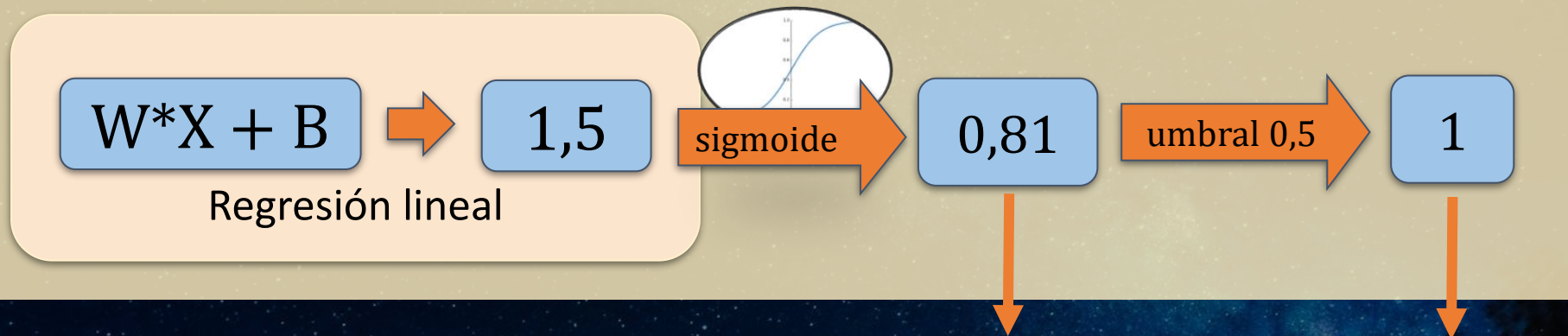


Regresión Logística Múltiple

The background of the slide is a deep blue night sky filled with numerous stars. A dark, horizontal band runs across the middle of the image, serving as a backdrop for the title text. The bottom of the image features the dark, silhouetted branches of evergreen trees, adding a naturalistic touch to the cosmic theme.

Regresión Logística (Binaria)



X					Y
mean radius	mean texture	worst concave points	worst symmetry	worst fractal dimensio	target
17,99	10,38	0,265	0,460	0,11	1
20,57	17,77	0,186	0,275	0,089	1
19,69	21,25	0,243	0,361	0,087	1
11,42	20,38	0,257	0,663	0,173	1
20,29	14,34	0,162	0,236	0,076	1
13,54	14,36	0,128	0,297	0,072	0
13,08	15,71	0,0728	0,318	0,081	0
9,504	12,44	0,0622	0,245	0,077	0
15,34	14,26	0,239	0,4667	0,099	1

Probabilidad de que $Y=1$
(tumor sea maligno)

Etiqueta predicha
por el modelo

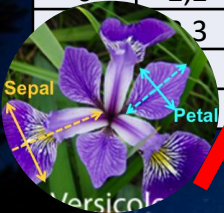
¿y si hay más clases?

Regresión Logística Múltiple

Clasificación Multilabel

Predicciones independientes

X				Y		
sepal_length	sepal_width	petal_length	petal_width	Setosa	Virginica	Versicolor
5	2	3,5	1	1	1	0
6	2,2	4	1	0	1	1
4,7	3,2	1,3	0,2	1	0	0
5,1	3,8	1,6	0,2	1	1	1
4,6	3,2	1,4	0,2	1	0	0
6,2	2,2	4,5	1,5	0	1	1
6	2,2	5	1,5	0	0	1
5,3	3	4	1,3	0	1	1
4,4	1,3	0	0	0	1	0
3,3	1	1	1	1	1	0



Clases mutuamente exclusivas

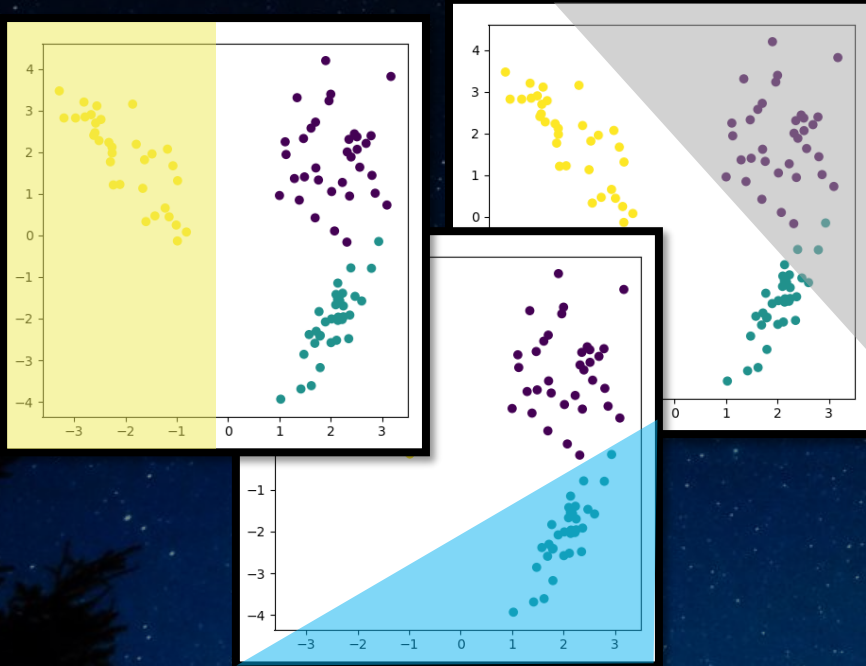
=> Problema de clasificación

X				Y		
sepal_length	sepal_width	petal_length	petal_width	Setosa	Virginica	Versicolor
5	2	3,5	1	0	1	0
6	2,2	4	1	0	1	0
4,7	3,2	1,3	0,2	1	0	0
5,1	3,8	1,6	0,2	1	0	0
4,6	3,2	1,4	0,2	1	0	0
6,2	2,2	4,5	1,5	0	1	0
6	2,2	5	1,5	0	0	1
5,5	2,3	4	1,3	0	1	0
6,3	2,3	4,4	1,3	0	1	0
5	2,3	3,3	1	0	1	0

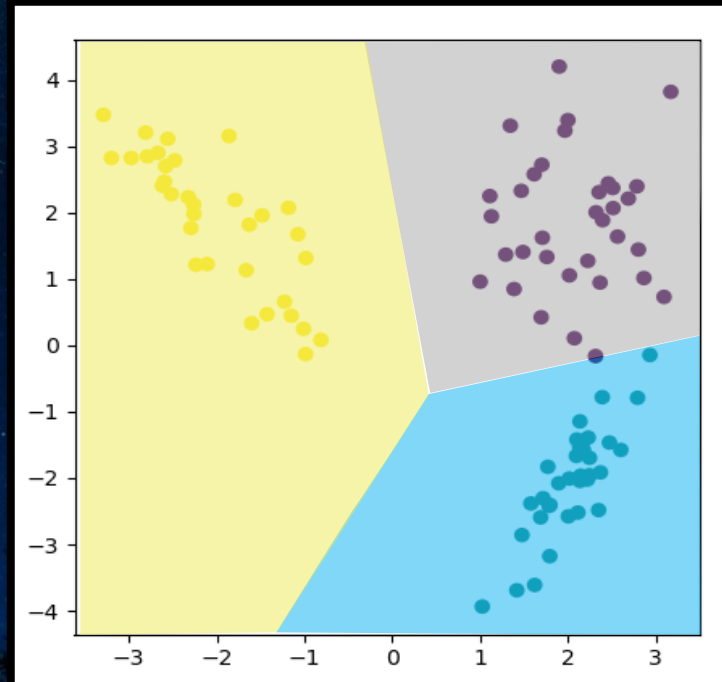
Regresión Logística Múltiple - OVR

One-vs-Rest

Un hiperplano por cada clase

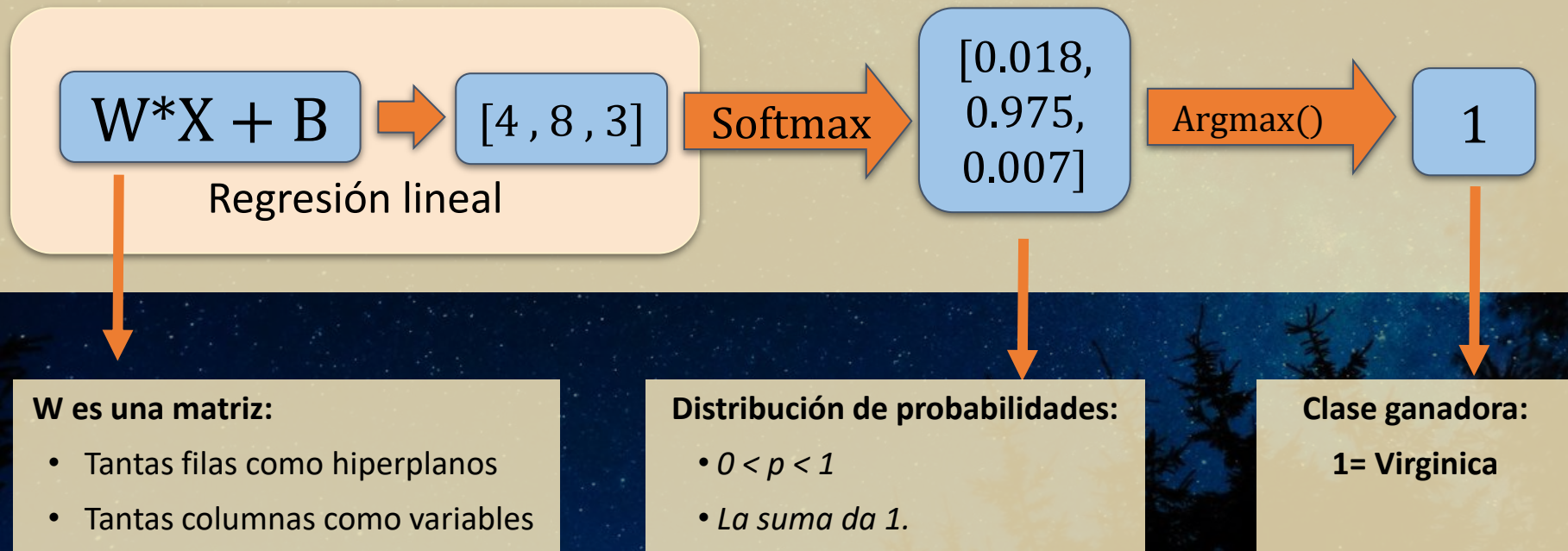


Fronteras de decisión del modelo



Regresión Logística Múltiple - Multinomial

Multinomial



Softmax

Función softmax

- Convierte salida de Regresión Lineal en distribución de probabilidades de c/ clase

1- Cálculo del vector E

$$E = [e^4, e^8, e^3] = [54.59, 2980.95, 20.08]$$

2- Cálculo del valor N

$$N = e^4 + e^8 + e^3 = 54.59 + 2980.95 + 20.08 = 3055.63$$

3- Cálculo del vector P de probabilidades

$$P = E / N = [e^4/N, e^8/N, e^3/N] \\ = [54.59/3055.63, 2980.95/3055.63, 20.08/3055.63]$$

[4, 8, 3]

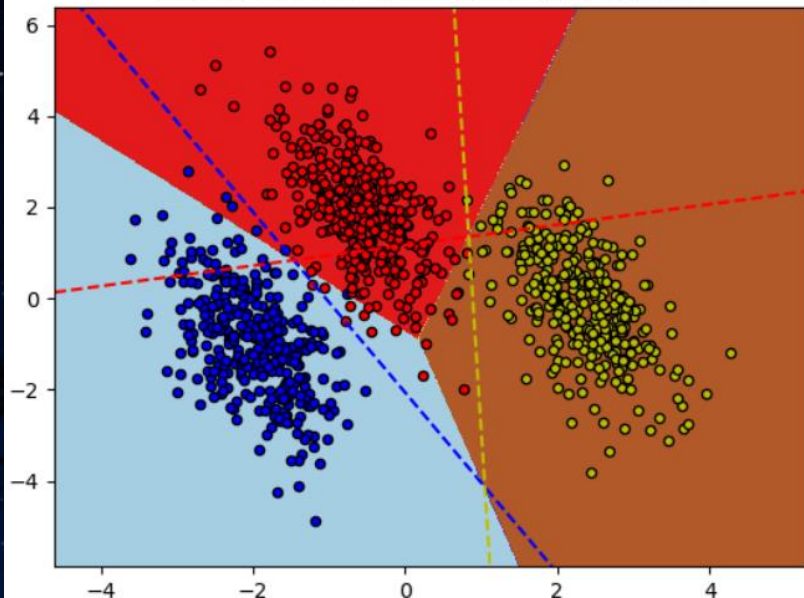
Softmax

[0.018,
0.975,
0.007]

OVR vs Multinomial

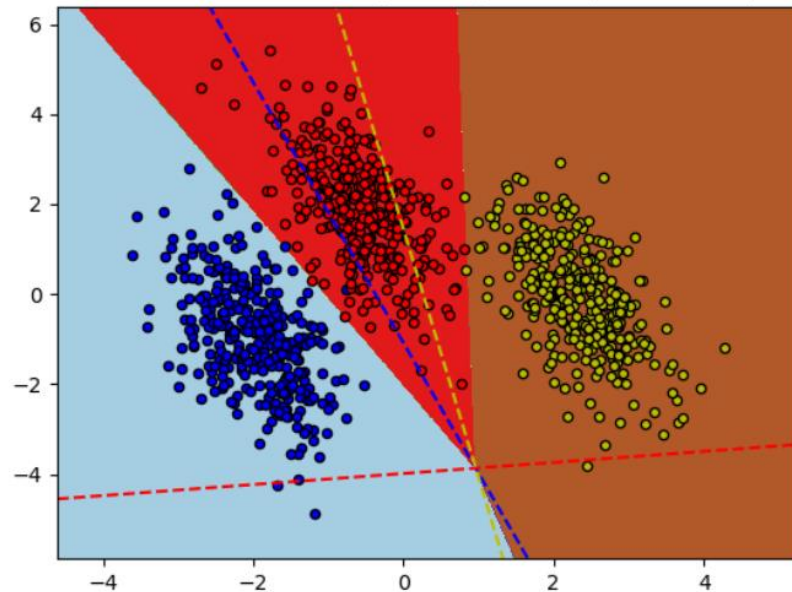
Generalmente para Binario

Decision surface of LogisticRegression (ovr)



El más utilizado para multiclase

Decision surface of LogisticRegression (multinomial)



Ejemplo: Iris dataset

```
df= pd.read_csv('iris.csv')  
x= df.drop('name', axis=1)  
y= df['name']
```

```
modelo= LogisticRegression()  
modelo.fit(x, y)
```

```
w= modelo.coef_
```

```
b= modelo.intercept_
```

```
y_pred= modelo.predict(x)
```

```
y_probs= modelo.predict_proba(x)
```

```
Acc= modelo.score(x,y)
```

	0	1	2	3
0	0.370203	1.37106	-2.20521	-1.00335
1	0.221242	-1.50411	0.572713	-1.24418
2	-1.50496	-1.36222	2.37392	2.4321

	0
0	0.621432
1	1.72006
2	-2.30145

X				Y
sepal_ length	sepal_ width	petal_ length	petal_ width	name
5	2	3,5	1	1
6	2,2	4	1	1
4,7	3,2	1,3	0,2	0
5,1	3,8	1,6	0,2	0
4,6	3,2	1,4	0,2	0
6,2	2,2	4,5	1,5	1
6	2,2	5	1,5	2
5,5	2,3	4	1,3	1
6,3	2,3	4,4	1,3	1
5	2,3	3,3	1	1

Ejemplo: Iris dataset

```
df= pd.read_csv('iris.csv')  
x= df.drop('name', axis=1)  
y= df['name']
```

```
modelo= LogisticRegression()  
modelo.fit(x, y)
```

```
w= modelo.coef_
```

```
b= modelo.intercept_
```

```
y_pred= modelo.predict(x)
```

```
y_probs= modelo.predict_proba(x)
```

```
Acc= modelo.score(x,y)
```

Etiquetas

0	2
1	1
2	2
3	2
4	1
5	2
6	2
7	0
8	0

Probabilidades

	0	1	2
0	0.000203437	0.40739	0.592406
1	0.0612542	0.661763	0.276983
2	0.000729856	0.133338	0.865932
3	0.000786158	0.225037	0.774177
4	0.012678	0.583475	0.403847
5	0.00399002	0.324424	0.671586
6	0.0120004	0.386509	0.60149
7	0.895409	0.104581	9.96441e-06
8	0.793618	0.206285	9.67433e-05

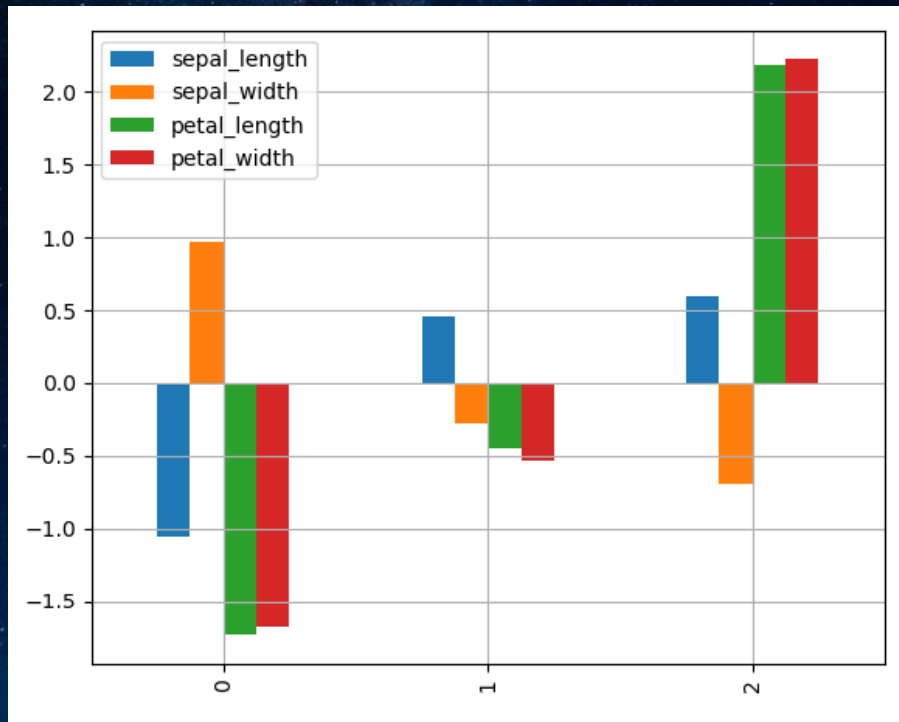
Accuracy

Ejemplo: Iris dataset

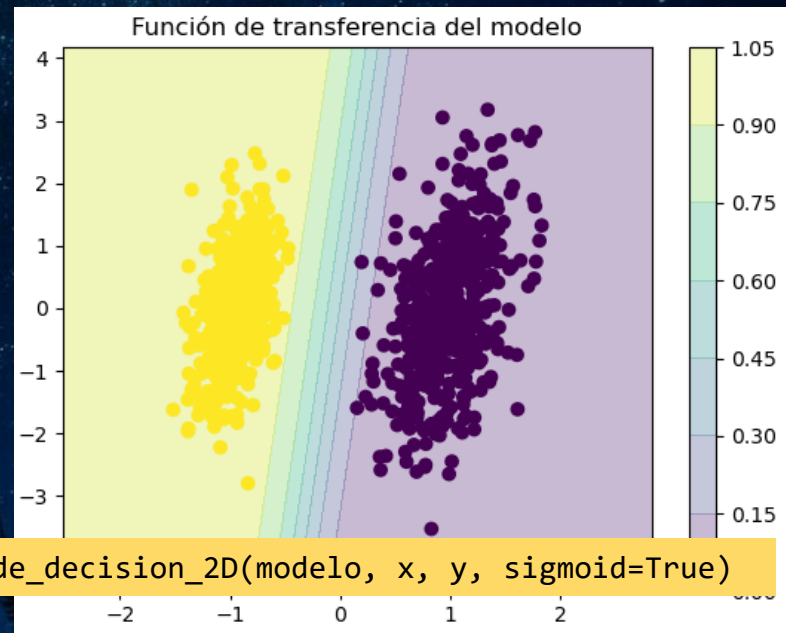
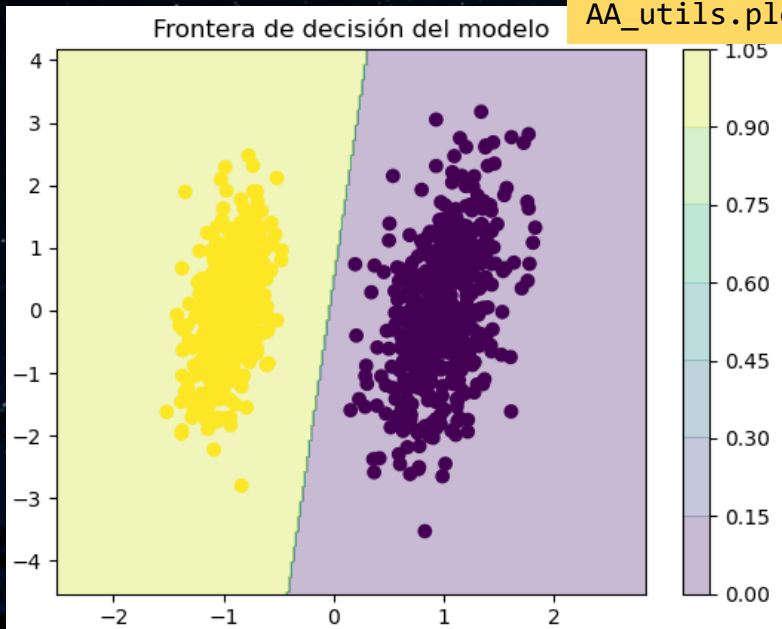
```
df2 = pd.DataFrame(w, columns=df.columns[0:-1])  
df2.plot(kind='bar'); plt.grid()
```

Coeficientes por cada clase (0,1,2)

Variables normalizadas



Graficar frontera de decisión 2D



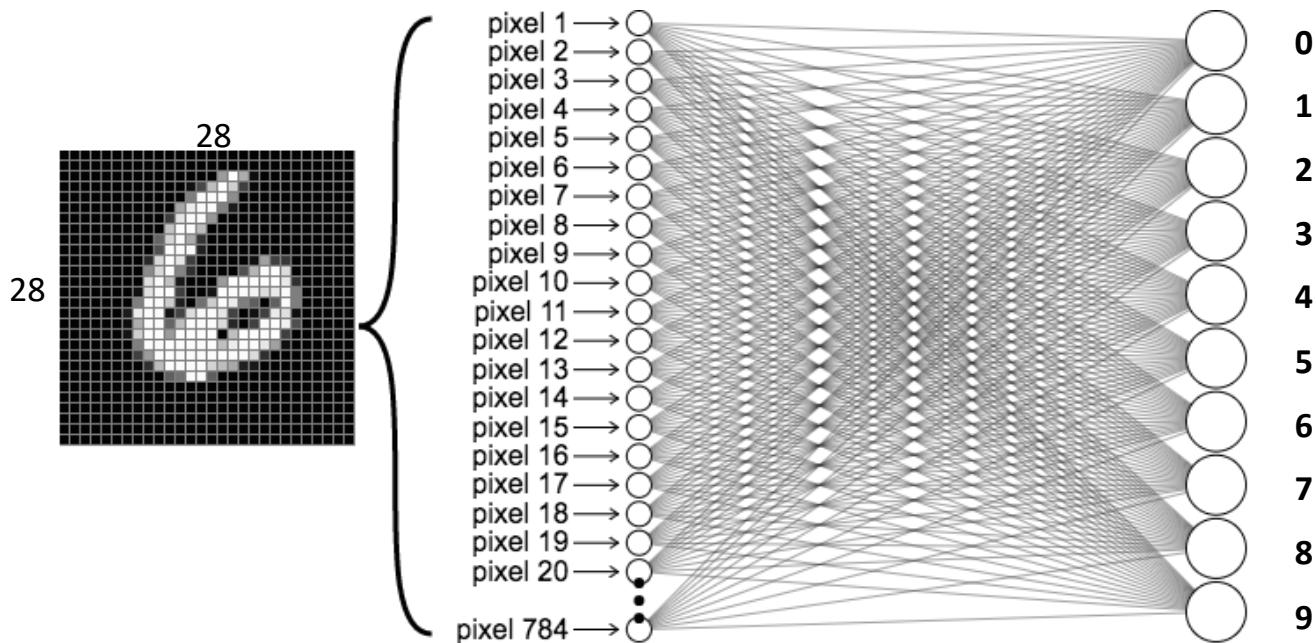
Clasificación de imágenes - MNIST

MNIST es una base de datos con decenas de miles de dígitos escritos a mano. El objetivo es entrenar un modelo que clasifique correctamente estos números. Cada dígito está codificado en 28x28 píxeles en escala de grises.



Clasificación de imágenes - MNIST

Aplanamos la imagen en un solo vector y lo alimentamos al Regresor Logístico.



Clasificación de imágenes - MNIST

Vector de pesos aprendidos por cada clase graficados en forma de matriz.
("Regresion_Logistica_MNIST.py")

```
w_r= np.reshape(w, (10, 28,28))
```

Coefficientes aprendidos para cada clase,
agrupados en forma matricial de 28x28

