



Semilleros
by DSRP

Ruta de Aprendizaje



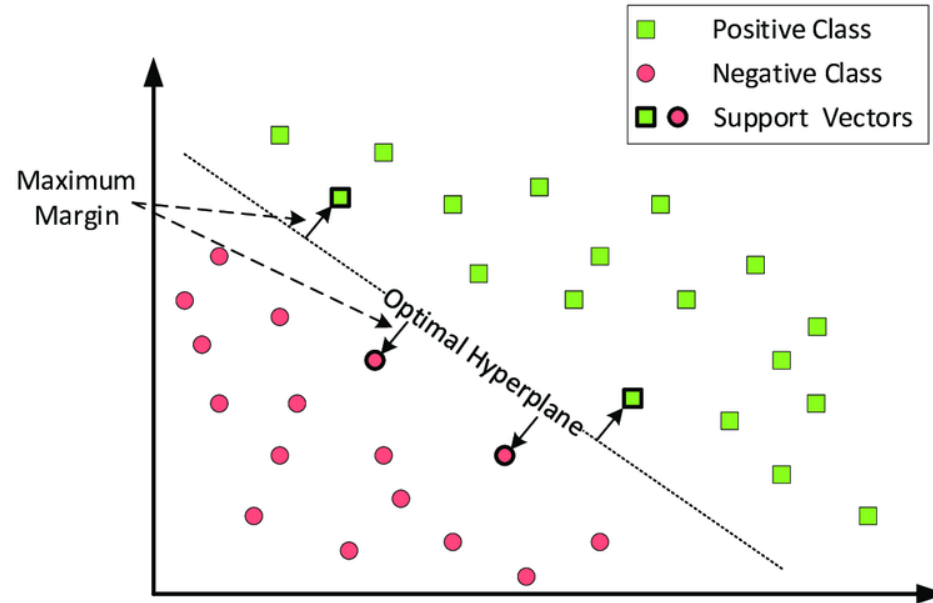
COMPUTER VISION

- 6 lecciones de Deep Learning

lesson

6

Binary Classification



Lesson 4.1: Binary Classification

Al momento de predecir:

Datos continuos

Datos discretos

La clasificación en una de dos clases es un problema común en Machine Learning. Es posible que desee predecir si es probable que un cliente realice una compra, si una transacción con tarjeta de crédito fue fraudulenta o no, una prueba médica de una enfermedad.

Todos estos son problemas de clasificación binaria.

```
[[0.99624395],  
 [1.         ],  
 [0.32295686],  
 [0.9996427 ],  
 [0.29272318],  
 [1.         ],  
 [0.29272318],  
 [0.9999912 ],  
 [0.99966645],  
 [1.         ],  
 [0.9999998 ],  
 [0.9999648 ],  
 [0.29272318],  
 [0.99952054],  
 [0.99982977],  
 [1.         ],
```

```
[1,  
 1,  
 0,  
 1,  
 0,  
 1,  
 0,  
 1,  
 1,  
 1,  
 1,  
 1,  
 0,  
 1,  
 1,  
 1,
```

En sus datos sin procesar, las clases pueden estar representadas por cadenas como "Sí" y "No", o "Puma" y "Gato". Antes de usar estos datos, asignaremos una etiqueta de clase: una clase será 0 y la otra será 1.

La asignación de etiquetas numéricas coloca los datos en una forma que una red neuronal puede usar.

Lesson 4.2: Accuracy and Cross-Entropy

Accuracy (Precisión)

Es una de las muchas métricas que se utilizan para medir el éxito en un problema de clasificación. **Accuracy** es la relación entre las predicciones correctas y las predicciones totales: **accuracy = number_correct/total**.

Un modelo que siempre predice correctamente tendría una puntuación de **accuracy** 1,0.

Cross Entropy (entropía cruzada)

El problema con el **accuracy** y con la mayoría de métricas de clasificación es que no se puede utilizar como función de pérdida. SGD necesita una función de pérdida que cambia suavemente, entonces necesitamos elegir un sustituto para que actúe como función de pérdida. Este sustituto es la función de pérdida de entropía cruzada (**Cross Entropy Loss**).

La entropía cruzada es una especie de medida de la distancia de una distribución de probabilidad a otra.

▶ #definir optimizador, función de pérdida y métricas

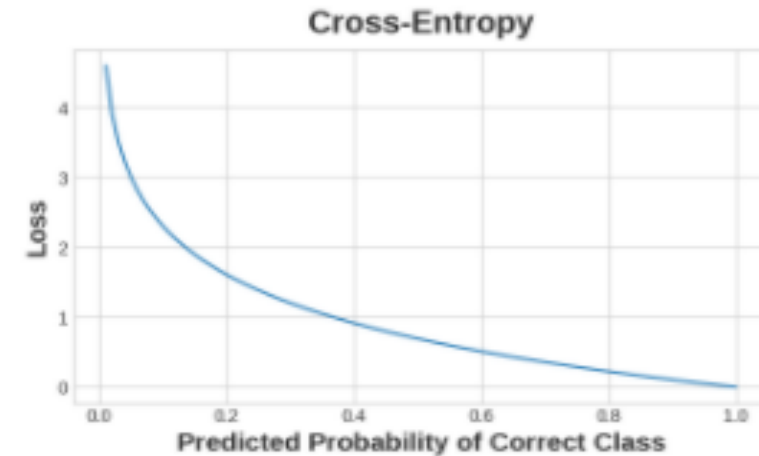
```
model.compile(  
    optimizer = 'adam',  
    loss = 'binary_crossentropy',  
    metrics=['binary_accuracy'],  
)
```

```
Epoch 173/188  
25/25 [=====] - 0s 3ms/step - loss: 0.0283 - binary_accuracy: 0.9973 - val_loss: 0.0380 - val_binary_accuracy: 1.0000  
Epoch 174/188  
25/25 [=====] - 0s 3ms/step - loss: 0.0281 - binary_accuracy: 0.9973 - val_loss: 0.0377 - val_binary_accuracy: 1.0000
```

Lesson 4.2: Accuracy and Cross-Entropy

La idea es que queremos que nuestra red prediga la clase correcta con probabilidad **1.0**. Cuanto más lejos esté la probabilidad predicha de 1.0, mayor será la pérdida de entropía cruzada.

Use la entropía cruzada para una pérdida de clasificación.

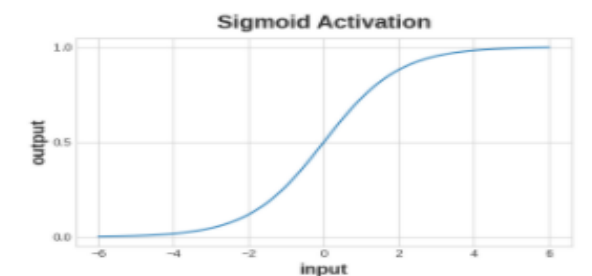


Cross-entropy penalizes incorrect probability predictions.

Lesson 4.3: Making probabilities with the Sigmoid Function

Las funciones de entropía cruzada y precisión requieren probabilidades como entradas, es decir, números del 0 al 1.

Para convertir las salidas de valor real producidas por una capa densa en probabilidades, adjuntamos un nuevo tipo de función de activación, la activación sigmoidea.



The sigmoid function maps real numbers into the interval $[0, 1]$.

Note: Por encima de 0.5 nos dará la clase correcta 1, y por debajo de 0.5 nos dará la clase incorrecta 0. Por ello se tiene que redondear si es necesario.

Función de activación "Sigmoide"

La función de activación sigmoide transforma los valores introducidos a una escala (0,1), donde los valores altos tienen de manera asintótica a 1 y los valores muy bajos tienden de manera asintótica a 0.

Características:

- Satura y mata el gradiente
- Lenta convergencia
- No esta centrada en el cero
- Esta acotada entre 0 y 1
- Buen rendimiento en la última capa

```
#crear capa de salida
```

```
layers.Dense(units=1, activation='sigmoid'),
])
```

```
[0.99624395],      [1,
 [1.          ],      1,
 [0.32295686],      0,
 [0.9996427 ],      1,
 [0.29272318],      0,
 [1.          ],      1,
 [0.29272318],      0,
 [0.9999912 ],      1,
 [0.99966645],      1,
 [          ],      1
```


LABORATORIO 6

BONUS LESSON!

Matriz de confusión y sus métricas

		Predicción			
		Positivo	Negativo		
Actual	Positivo	Verdaderos Positivos	Falsos Negativos	dato real = 1	dato predicho = 0
	Negativo	Falsos Positivos	Verdaderos Negativos	dato real = 0	dato predicho = 0

dato real = 1 dato real = 0
dato predicho = 1 dato predicho = 1

Lesson Bonus_1: Matriz de Confusión

Una matriz de confusión o también conocida matriz de error, es una tabla que sirve para evaluar el rendimiento de un modelo de clasificación. El número de predicciones correctas e incorrectas se resumen con los valores de conteo y se desglosan por cada clase.

		Actual Values		
		Yes	No	
Predicted Values	Yes	True Positive	False Positive	2x2
	No	False Negative	True Negative	

Falso Positivo: predijo que era positivo cuando en realidad era negativo.

Falso Negativo: predijo que era negativo cuando en realidad si era positivo.

Lesson Bonus_2: Métricas de la Matriz de Confusión - Accuracy

Prediction \ Class	True	False
True	True Positive	False Positive
False	False Negative	True Negative

Entre las principales métricas de una Matriz de confusión, podemos tener las siguientes:

Es la relación entre las predicciones correctas y las predicciones totales. Por lo tanto, es el cociente entre los casos bien clasificados por el modelo, y la suma de todos los casos.

Accuracy (exactitud)

Note: Cuando el dataset está desequilibrado, no es una métrica útil. Se recomienda utilizar la métrica **f1_score**

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}$$

TP = True Positive (Verdadero Positivo)

TN = True Negative (Verdadero Negativo)

FP = False Positive (Falso Positivo)

FN = False Negative (Falso Negativo)

Lesson Bonus_2: Métricas de la Matriz de Confusión - Accuracy

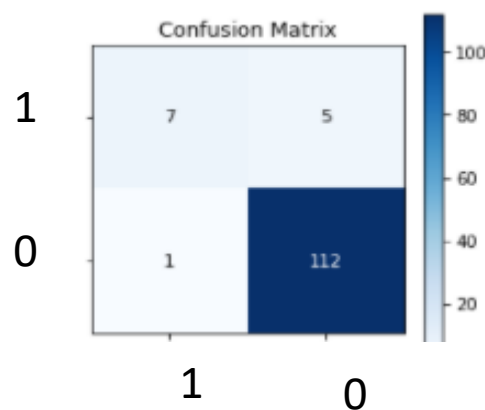
Prediction \ Class	True	False
True	True Positive	False Positive
False	False Negative	True Negative

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

TP = True Positive (Verdadero Positivo)
TN = True Negative (Verdadero Negativo)
FP = False Positive (Falso Positivo)
FN = False Negative (Falso Negativo)

```
from sklearn.metrics import accuracy_score
precisión = accuracy_score(val_y, y_pred1)
print(precisión)

!pip install -q scikit-plot
import scikitplot as skplt
skplt.metrics.plot_confusion_matrix(val_y, y_pred1, figsize=(4,4))
```

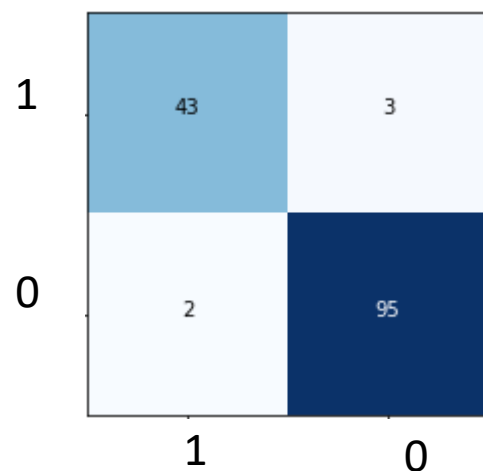


```
0.952
<matplotlib.axes._subplots.AxesSubplot at 0x7f0eb662b550>
```

$$\text{Accuracy} = (7 + 112) / (7 + 5 + 112 + 1) = (119 / 125) = 0.952$$

95.2 %

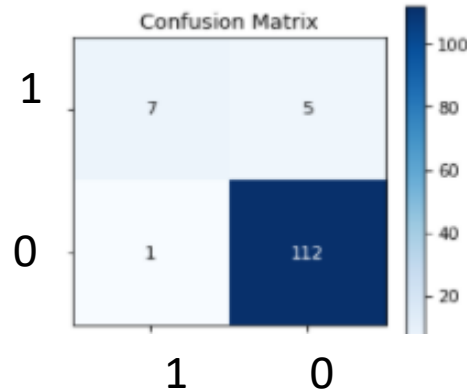
Ejercicio



Accuracy =

Lesson Bonus_2: Métricas de la Matriz de Confusión - Precision

Precision (Precisión)



La precisión también se conoce como valor predictivo positivo. Es la proporción de instancias relevantes entre las instancias recuperadas.

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Precision} = (7)/(7+5)$$

$$\text{Precision} = 58.3\%$$

Precisión es dengue

$$\text{Precision} = \frac{TN}{TN+FN}$$

$$\text{Precision} = (112)/(112+1)$$

$$\text{Precision} = 99.1\%$$

Precisión no es dengue

TP = True Positive (Verdadero Positivo)
TN = True Negative (Verdadero Negativo)
FP = False Positive (Falso Positivo)
FN = False Negative (Falso Negativo)

Lesson Bonus_2: Métricas de la Matriz de Confusión - Recall

Recall (sensibilidad)

Recall representa la tasa de verdaderos positivos. Es decir, es la proporción entre los casos positivos bien clasificados por el modelo, respecto al total de positivos.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

TP = True Positive (Verdadero Positivo)

TN = True Negative (Verdadero Negativo)

FP = False Positive (Falso Positivo)

FN = False Negative (Falso Negativo)

Lesson Bonus_2: Métricas de la Matriz de Confusión - Specificity

Specificity (Especificidad)

Recall representa la tasa de verdaderos negativos. Es decir, es la proporción entre los casos negativos bien clasificados por el modelo, respecto al total de negativos.

$$\text{Specificity} = \frac{TN}{TN + FP}$$

TP = True Positive (Verdadero Positivo)

TN = True Negative (Verdadero Negativo)

FP = False Positive (Falso Positivo)

FN = Falso Negative (Falso Negativo)

Métricas de la Matriz de Confusión – F1 score

F1 score (Puntuación F1)

F1 score es una métrica muy empleada porque nos resume la **Precision** y **Recall** en una sola métrica. Por ello, es de gran utilidad cuando la distribución de clases es desigual (desquilibrada).

Por ejemplo: en un dataset, tenemos 85% de datos positivos y un 15% de datos negativos, lo que en el campo de salud es bastante común.

$$F1\ score = \frac{2 * (precision * recall)}{precision + recall} = \frac{2TP}{2TP + FP + FN}$$

TP = True Positive (Verdadero Positivo)

TN = True Negative (Verdadero Negativo)

FP = False Positive (Falso Positivo)

FN = False Negative (Falso Negativo)

```
# Method 1: sklearn
from sklearn.metrics import f1_score
f1_score(y_true, y_pred, average=None)
# Method 2: Manual Calculation
F1 = 2 * (precision * recall) / (precision + recall)
# Method 3: Classification report [BONUS]
from sklearn.metrics import classification_report
print(classification_report(y_true, y_pred, target_names=target_names))
```