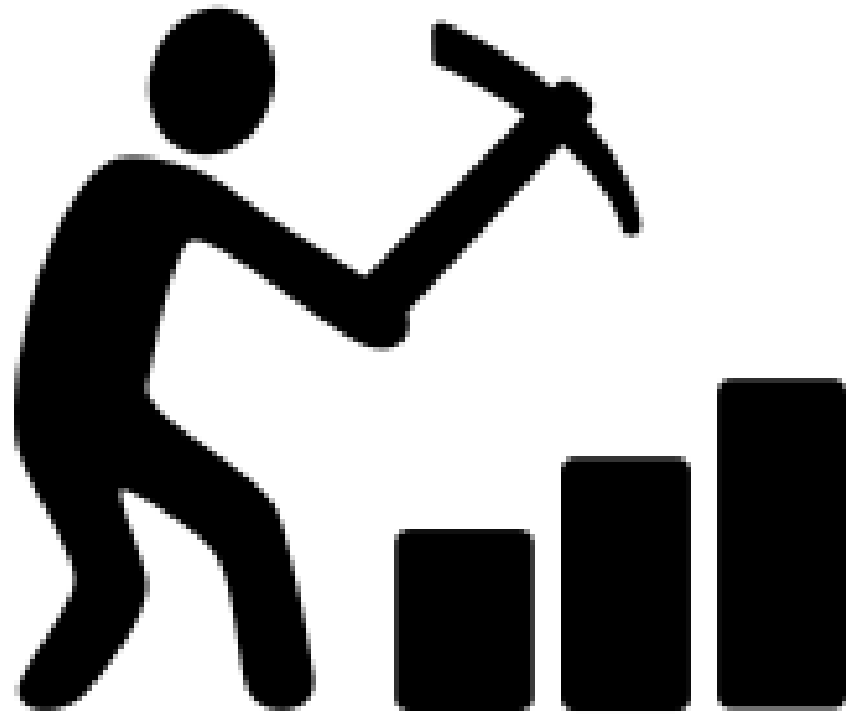


# Métricas de Evaluación

Sofia Paola Salazar Coronado	1679700
Andrés Chávez Peña	1941534
Ricardo Emilio Lemus Morales	1852600



# 1 ¿Para qué son las métricas de evaluación?

Se usan para medir y evaluar la calidad del estudio . Al tener una sola medida se limita la visión del modelo, por ende se checan múltiples medidas.

2

**"What do we want  
to optimize for?"**





## 5 Métodos

1. Accuracy, Precision, and Recall
2. F1 Score
3. AUC/ROC
4. Log Loss/Binary Crossentropy
5. Categorical Crossentropy



# I Puntería, Precisión y recordar

(Accuracy, Precision, and Recall)

<https://towardsdatascience.com/the-5-classification-evaluation-metrics-you-must-know-aa97784ff226>

# I Puntería, Precisión y recordar

## Verdadero positivo (VP):

- Realidad: Maligno
- Predicción del modelo de AA: Maligno
- Número de resultados de VP: 1

## Falso positivo (FP):

- Realidad: Benigno
- Predicción del modelo de AA: Maligno
- Número de resultados de FP: 1

## Falso negativo (FN):

- Realidad: Maligno
- Predicción del modelo de AA: Benigno
- Número de resultados de FN: 8

## Verdadero negativo (VN):

- Realidad: Benigno
- Predicción del modelo de AA: Benigno
- Número de resultados de VN: 90

$$\text{Exactitud} = \frac{VP + VN}{VP + VN + FP + FN} = \frac{1 + 90}{1 + 90 + 1 + 8} = 0.91$$

Esta métrica está dada como el porcentaje de las predicciones correctas que se hicieron sobre el total de intentos realizados

<https://developers.google.com/machine-learning/crash-course/classification/accuracy>

# I Puntería, Precisión y recordar

```
[[10745    173]
 [   261    442]]
```

Con los datos ya podemos utilizar sacar nuestras métricas

```
In [2]: from sklearn.metrics import accuracy_score
print("Accuracy : %.2f"%(accuracy_score(y_test, predictions)*100))
```

Accuracy : 96.27

Accuracy

```
In [4]: from sklearn.metrics import recall_score
print("Recall : %.2f"%(recall_score(y_test, predictions)*100))
```

Recall : 62.87

Precision

```
In [6]: from sklearn.metrics import precision_score
print("Precision : %.2f"%(precision_score(y_test, predictions)*100))
```

Precision : 71.87



# II Puntuación F1

.(F1 Score).

<https://www.pluralsight.com/guides/evaluating-a-data-mining-model>



## II Puntuación F1



$$F1 = \frac{2 * precision * recall}{precision + recall}$$

$$F1 = \frac{2 \times 0.3 \times 0.1}{0.3 + 0.1} \therefore F1 = 0.15$$

Harmonic mean is conservative mean compared to Arithmetic mean and geometric mean.  
It means that Harmonic mean is nearest to the smallest of the input numbers.

Cuando “Recordar” y “Precisión” son necesarias se utiliza. Esta trata de balancear ambas métricas. Matemáticamente hablando esta definida como la media armónica de “Recordar” y “Precisión”.

## II Puntuación F1

### **F1 test**

Con nuestros datos ya podemos hacer el F1 TEST

```
In [8]: from sklearn.metrics import f1_score  
print("F1 Score : %.2f"%(f1_score(y_test, predictions)*100))
```

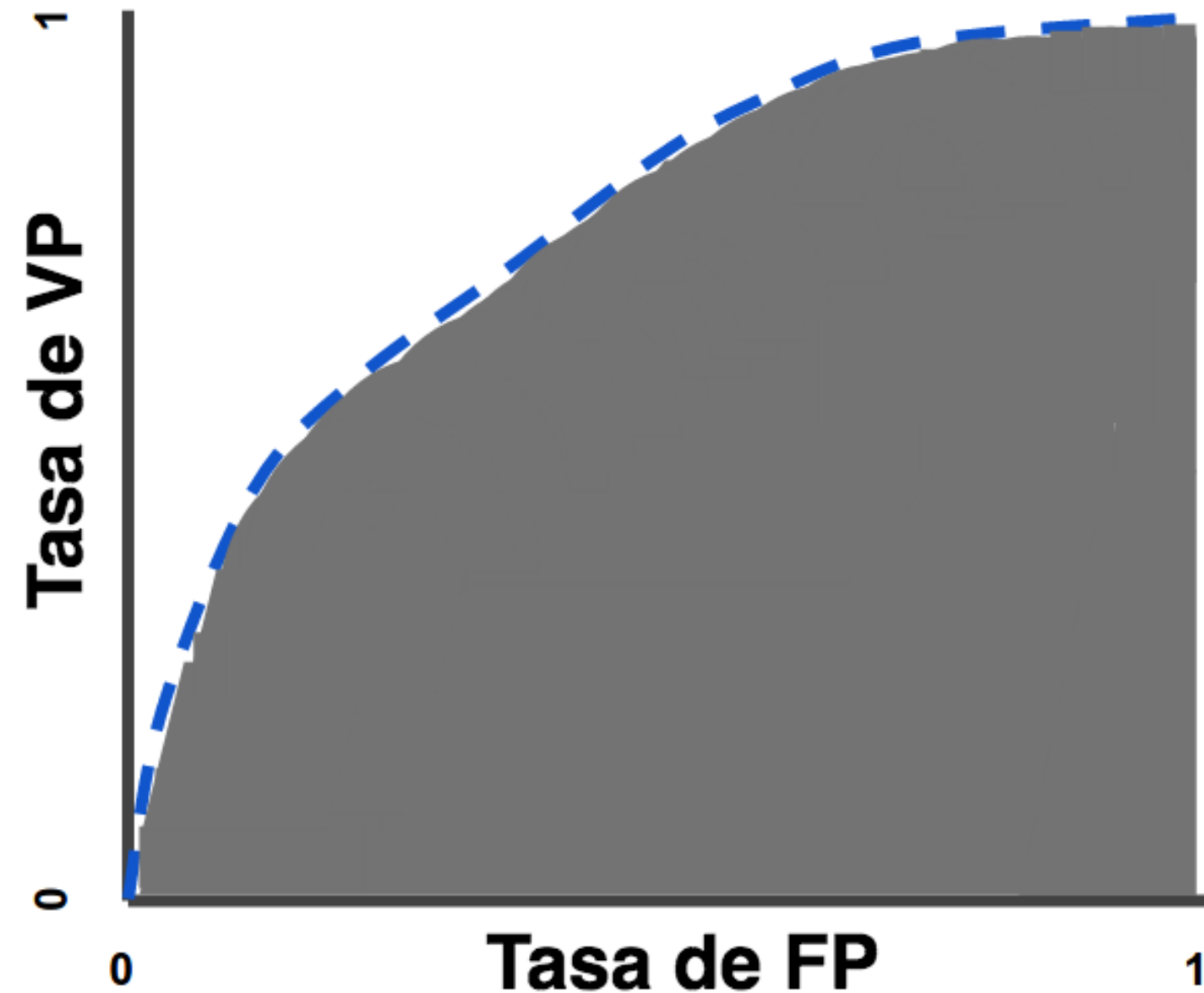
F1 Score : 67.07

# III AUC/ROC

.(AUC/ROC).  
.

<https://www.pluralsight.com/guides/evaluating-a-data-mining-model>

# III AUC/ROC



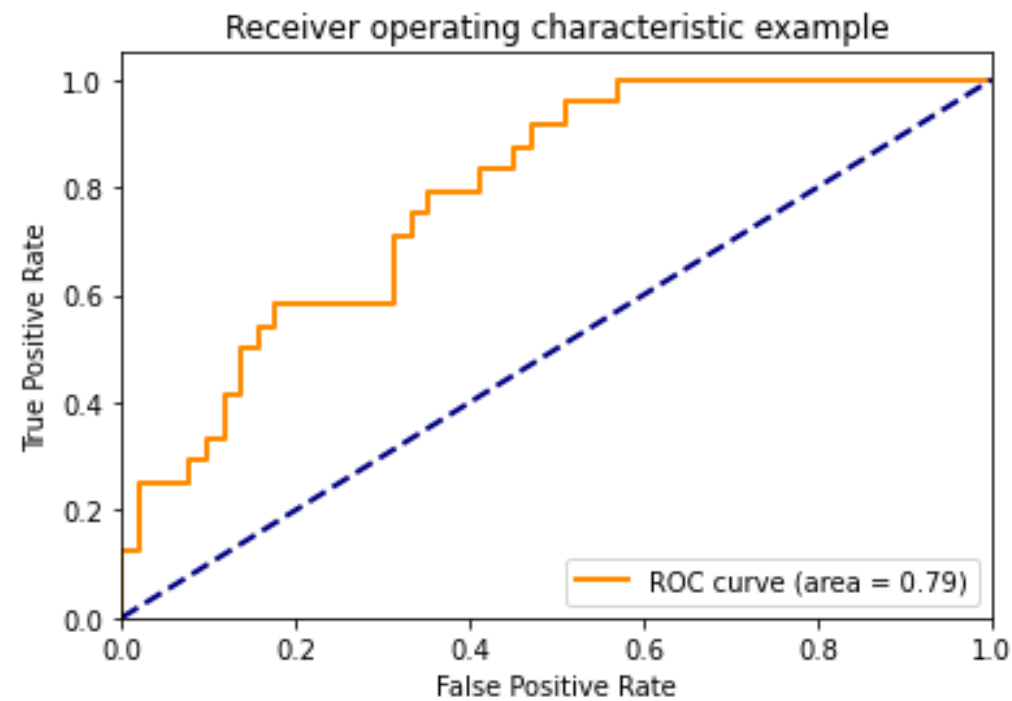
Curva ROC - Mide qué tan bien se clasifican las predicciones en un gráfico que muestra el rendimiento de un modelo de clasificación en todos los umbrales de la misma.

Curva AUC- Mide la calidad de las predicciones del modelo ,significa "área bajo la curva ROC".



# III AUC/ROC

```
In [14]: plt.figure()
lw = 2
plt.plot(fpr[2], tpr[2], color='darkorange',
         lw=lw, label='ROC curve (area = %0.2f)' % roc_auc[2])
plt.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic example')
plt.legend(loc="lower right")
plt.show()
```

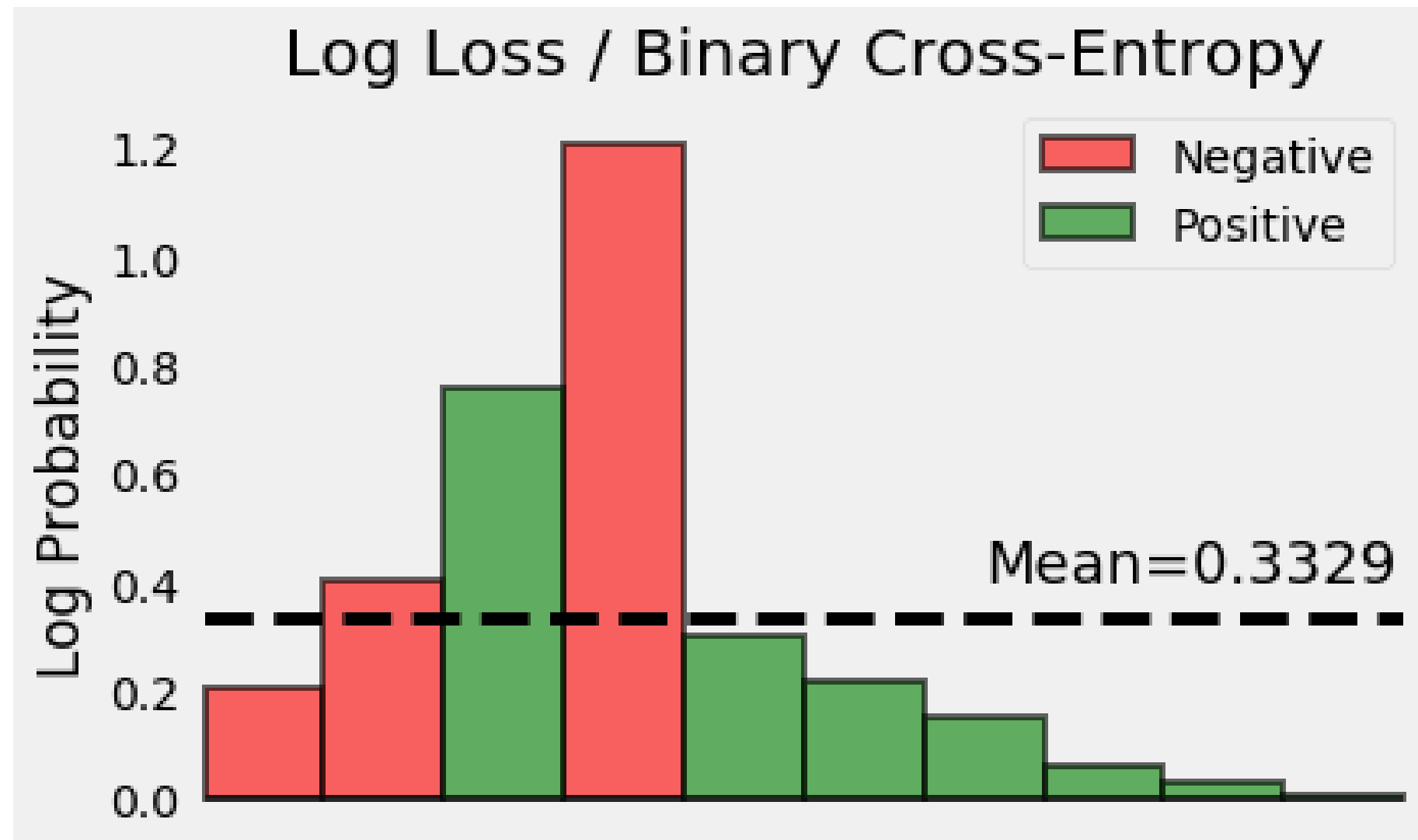


# IV Perdida logística y entropía cruzada

.(Log\_Loss/Binary\_Crossentropy).

<https://www.pluralsight.com/guides/evaluating-a-data-mining-model>

# IV Pérdida logística y entropía cruzada



Entropía cruzada (cross over entre probabilidades)

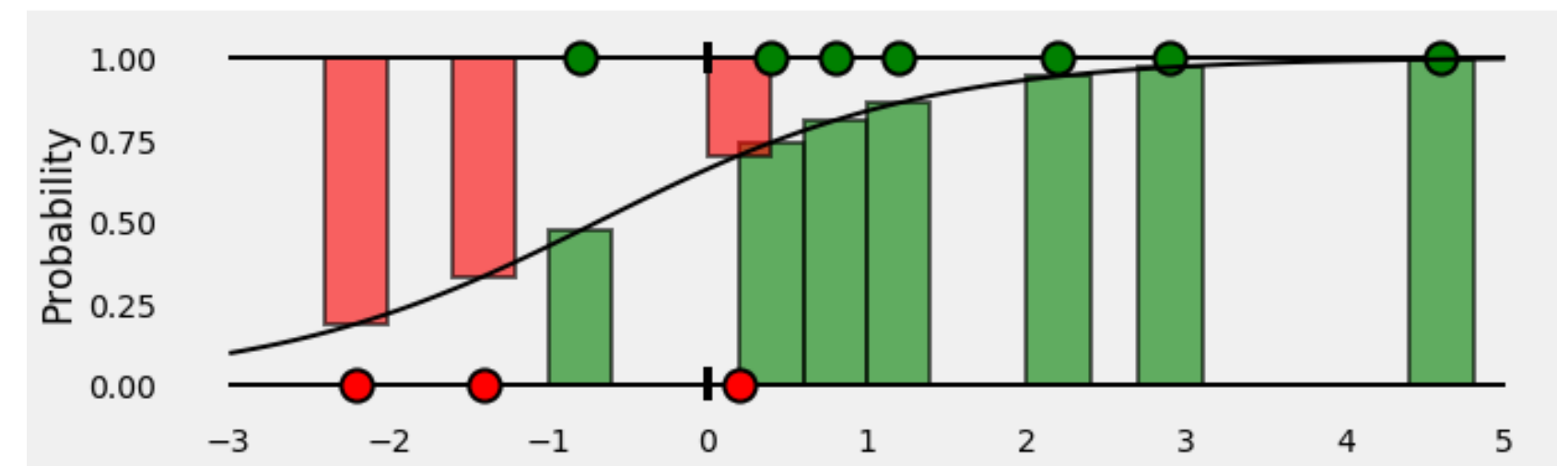
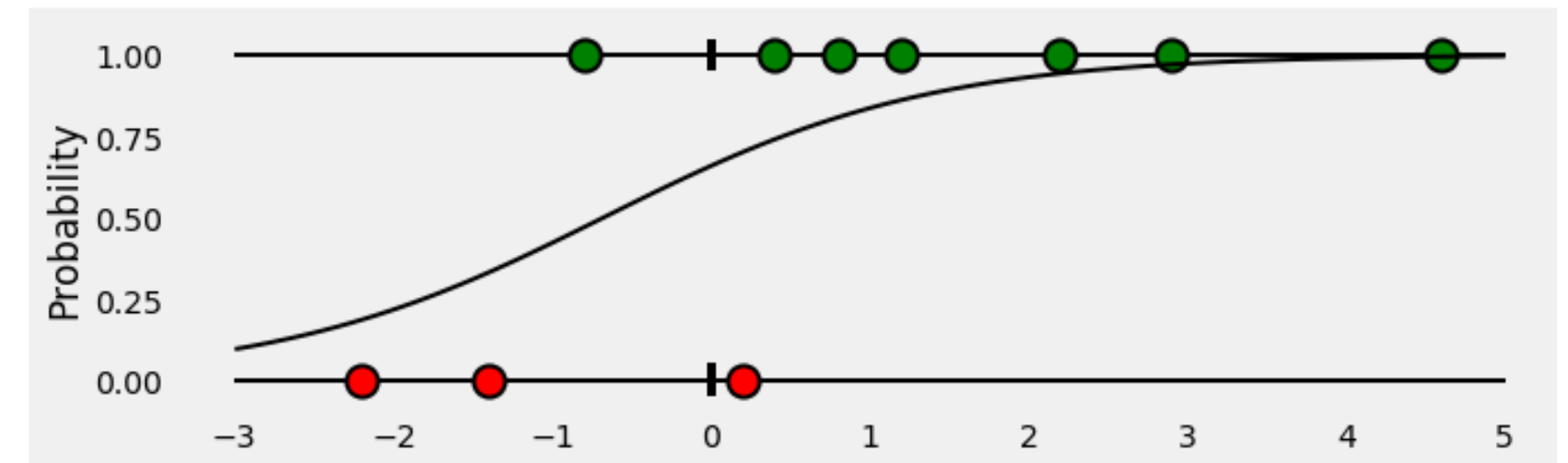
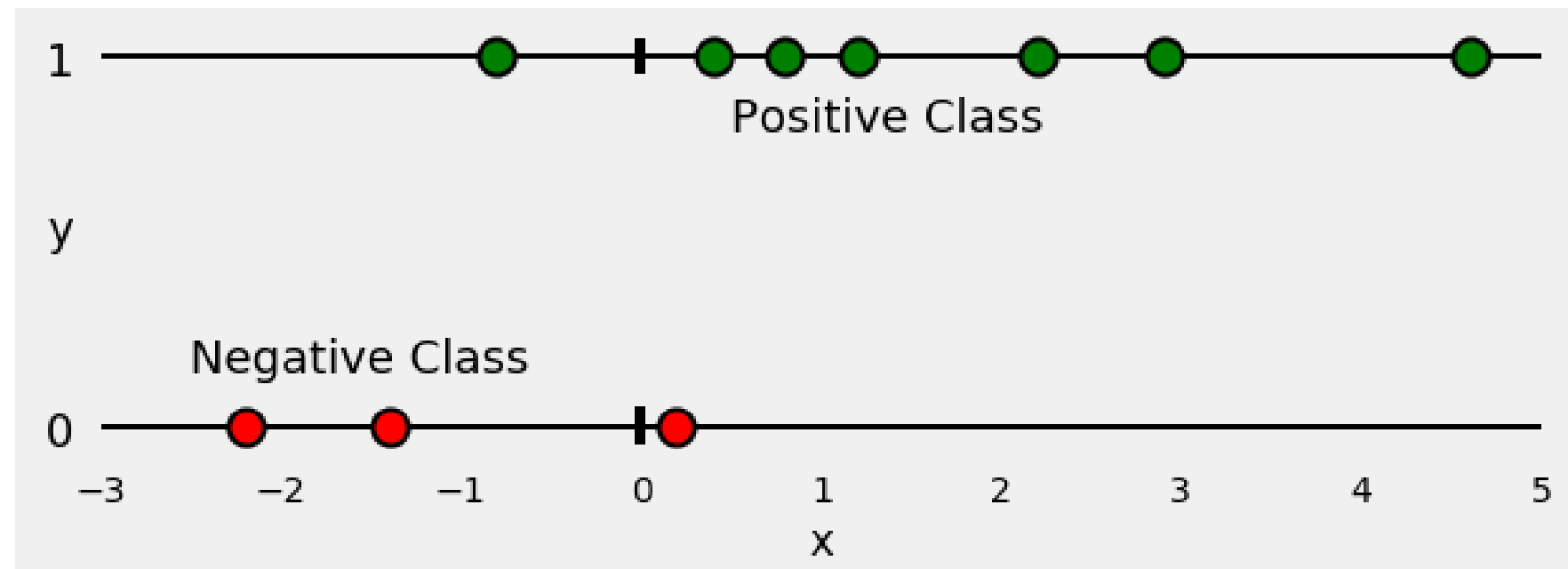
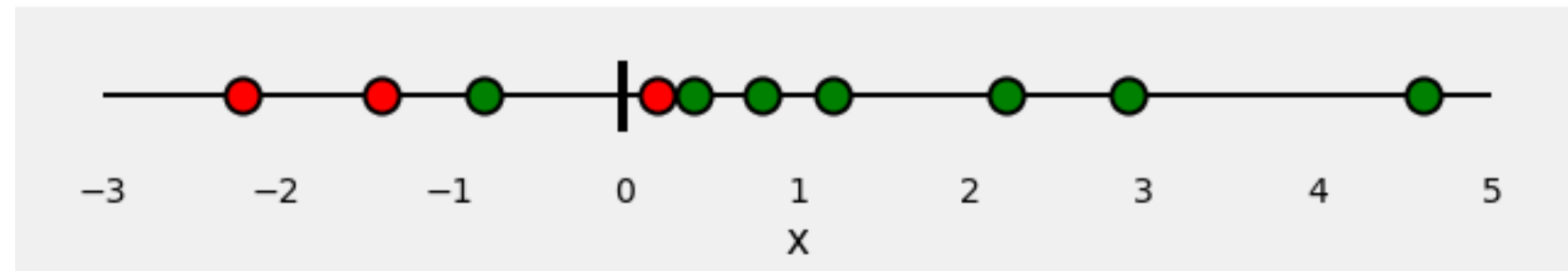
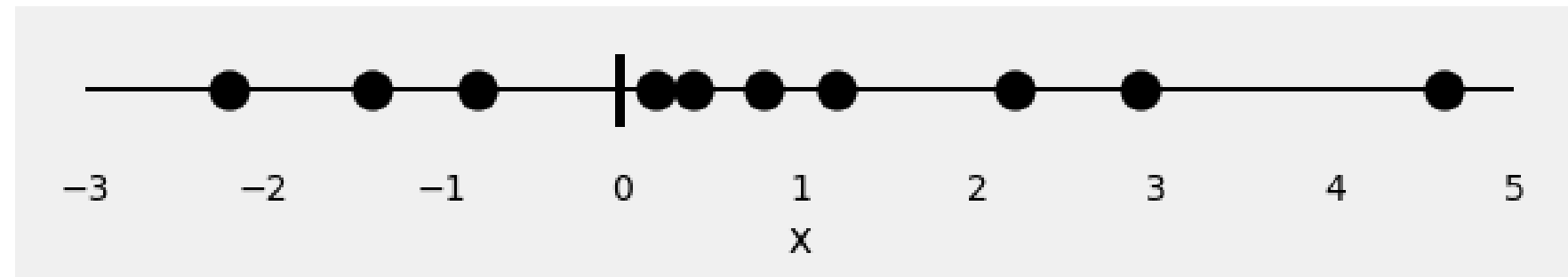
Es la métrica más importante al analizar probabilidades, un menor valor "log-loss" representa una mejor predicción

La entropía cruzada entre dos distribuciones de probabilidad mide la media de bits necesarios para identificar un evento de un conjunto de posibilidades

$$H_p(q) = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))$$

<https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a>

# IV Pérdida logística y entropía cruzada



<https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a>



# IV Perdida logística y entropía cruzada

## Log Loss

```
In [15]: from sklearn.linear_model import LogisticRegression
from sklearn.metrics import log_loss
import numpy as np

x = np.array([-2.2, -1.4, -.8, .2, .4, .8, 1.2, 2.2, 2.9, 4.6])
y = np.array([0.0, 0.0, 1.0, 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0])

logr = LogisticRegression(solver='lbfgs')
logr.fit(x.reshape(-1, 1), y)

y_pred = logr.predict_proba(x.reshape(-1, 1))[:, 1].ravel()
loss = log_loss(y, y_pred)

print('x = {}'.format(x))
print('y = {}'.format(y))
print('p(y) = {}'.format(np.round(y_pred, 2)))
print('Log Loss / Cross Entropy = {:.4f}'.format(loss))

x = [-2.2 -1.4 -0.8  0.2  0.4  0.8  1.2  2.2  2.9  4.6]
y = [0.  0.  1.  0.  1.  1.  1.  1.  1.  1.]
p(y) = [0.19 0.33 0.47 0.7  0.74 0.81 0.86 0.94 0.97 0.99]
Log Loss / Cross Entropy = 0.3329
```

<https://github.com/Emilio741/Mineria-de-Datos/blob/main/Ejemplos%20Python.ipynb>

# V Entropía cruzada categórica

(Categorical Crossentropy)

<https://www.pluralsight.com/guides/evaluating-a-data-mining-model>

# V Entropía cruzada categorica

Target feature Encoding Categorical
Tiger
Cat
Airplane
Cat

Encoding	Tiger ↓	Cat ↓	Airplane ↓
→	1	0	0
→	0	1	0
→	0	0	1
→	0	1	0

Se utiliza para medir categorías, como por ejemplo la reseña final de una película.

$$\text{LogarithmicLoss} = \frac{-1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{ij} * \log(p_{ij})$$

<https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/categorical-crossentropy>.

[https://gombru.github.io/2018/05/23/cross\\_entropy\\_loss/](https://gombru.github.io/2018/05/23/cross_entropy_loss/)