XUNTA DE GALICIA | CENTRO DE FORMACIÓN E RECURSOS DE FERROL

Inteligencia Artificial para la Sociedad

Alma Mallo (UDC)

Francisco Bellas (UDC)

Diciembre de 2024

Organización del curso

1. Sesión 1: introducción a la IA y aspectos ético-legales
2. Sesión 2: percepción y actuación
3. Sesión 3: representación y razonamiento
4. Sesión 4: aprendizaje automático 1
5. **Sesión 5: aprendizaje automático 2**
6. Sesión 6: herramientas de IA para la educación (IA generativa)

Sesión 5:
aprendizaje automático

Aprendizaje por refuerzo

- El aprendizaje por refuerzo es un área del aprendizaje automático inspirada en la psicología del comportamiento.
- Consiste en determinar qué acciones debe elegir un agente informático en un entorno dado para maximizar cierta noción de "recompensa".
- El objetivo del aprendizaje por refuerzo consiste en encontrar una política π que, para cada estado en el que se encuentre el agente, indique la mejor acción que puede emprenderse para alcanzar un objetivo determinado.

Aprendizaje por refuerzo

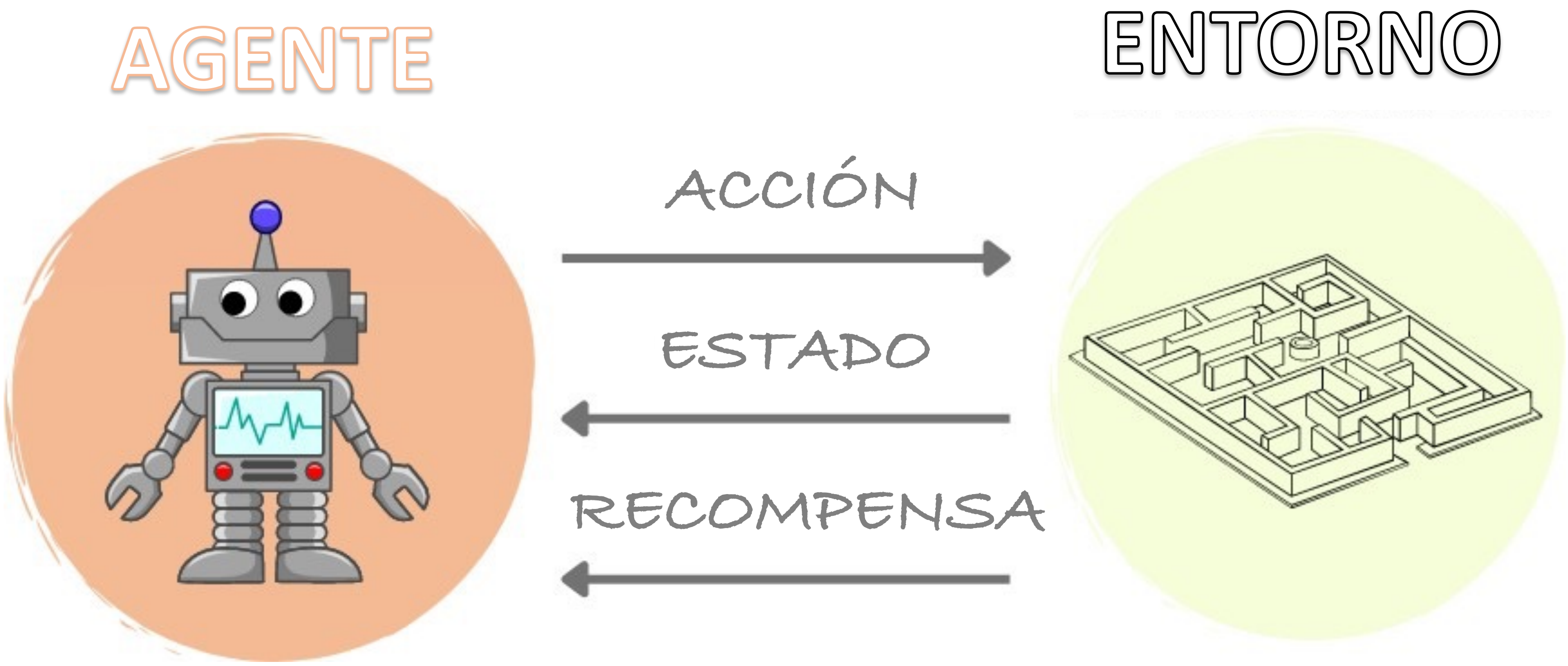
Este tipo de aprendizaje debe tener los dos componentes siguientes:

- Agente: será nuestro modelo que queremos entrenar y aprender a tomar decisiones.
- Entorno: será donde el agente interactúe y se "mueva". El entorno contiene las restricciones y reglas posibles en cada momento.

Además, existe una relación entre ellos que se retroalimenta y tiene los siguientes vínculos:

- Acción: las posibles acciones que puede realizar en un momento dado el agente.
- Estado: son los indicadores del entorno de cómo se encuentran en ese momento los distintos elementos que lo componen.
- Recompensas: como resultado de cada acción realizada por el agente, podemos obtener una recompensa o una penalización que orientará al agente sobre si lo está haciendo bien o mal.

Aprendizaje por refuerzo



Aprendizaje por refuerzo

- El agente, se encuentra en un estado inicial y realiza una acción que influye en el entorno. Esta decisión tendrá consecuencias: en la siguiente iteración, el agente se encontrará en un nuevo estado y obtendrá una recompensa.
  - Si la recompensa es positiva, estaremos reforzando ese comportamiento para el futuro. Esto ocurre cuando la acción realizada ha sido la correcta. En cambio, si la recompensa es negativa, estaremos penalizando al agente para que, ante la misma situación, actúe de forma diferente.
- Al principio, el agente no sabe nada sobre qué hacer o cómo comportarse. Así que que selecciona las acciones posibles al azar. Y obtendrá pistas sobre si lo está haciendo bien o mal en función de las recompensas que vaya obteniendo.

Aprendizaje por refuerzo: Q-learning

- Se trata de encontrar la siguiente mejor acción, dado un estado actual, tratando de maximizar la recompensa.

- El objetivo principal a la hora de entrenar nuestro modelo a través de las simulaciones será "rellenar" una tabla de políticas para que las decisiones que tome nuestro agente obtengan "la mayor recompensa" a la vez que avanzamos y no nos quedamos estancados, es decir, poder cumplir el objetivo global (o final) que queremos conseguir.

Aprendizaje por refuerzo: Q-learning

- En este tipo de modelo, es necesario conocer los siguientes conceptos:
  - Q-table: la tabla de políticas. Se rellena automáticamente utilizando el algoritmo de aprendizaje durante la fase de aprendizaje. El objetivo es aprender la tabla Q.
  - Valores Q: se utilizan para determinar lo buena que es una Acción, A, realizada en un estado concreto, S. → Q (A, S).
- Funciona en dos fases:
  - Aprendizaje: se actualizarán los valores de la tabla Q.
  - Ejecución: se selecciona la acción con mayor valor en la tabla Q en el estado en el que se encuentra.

Aprendizaje por refuerzo: Q-learning

- La tabla Q tiene la forma siguiente:

Q(s,a)	A1	A2	A3	…	An
S0					
S1					
S2					
…					
Sn					

Aprendizaje por refuerzo: Q-learning

- Durante la fase de aprendizaje, la selección de la acción a realizar se realiza de dos formas:
  - Selección aleatoria: permite al agente conocer el resultado de las acciones, ya que todavía no sabe cual es mejor. Se establece un porcentaje de exploración para esta selección aleatoria.
  - Selección de la mejor acción: mayor valor de la tabla Q para el estado del robot.
- Tras realizar la acción seleccionada, se calcula la recompensa y se actualiza la tabla Q.

Aprendizaje por refuerzo: Q-learning

- La actualización de la tabla Q se realiza aplicando la fórmula:

$$Q(s_t, a) = (1 - \propto) \cdot Q(s_t, a) + \propto \cdot [R(s_t, a) + \ \gamma \max Q(s_{t+1}, a)]$$

  - $Q(s_t, a)$: Antiguo valor almacenado en la tabla Q para ese estado y acción.
  - $\propto$: tasa de aprendizaje. Valor entre 0 y 1. Valores cercanos a 1 tiene mucho más en cuenta las nuevas acciones.
  - γ: factor de descuento. Valor entre 0 y 1. Valores cercanos a 0, tiene en cuenta los valores inmediatos.
  - $R(s_t, a)$: recompensa obtenida por ejecutar la acción *a* en el estado *s*.
  - $\max Q(s_{t+1}, a)$: máximo valor de recompensa que se obtendrá al transitar al estado s. Es el valor máximo de la tabla para el estado s.

Actividad: Aprendizaje por refuerzo

- **Objetivo**: implementar un programa que permita a Robobo aprender de forma autónoma a moverse en un laberinto.
- Para ello, se utilizará el algoritmo de aprendizaje por refuerzo Q-Learning.
- El primer paso es determinar los estados en los que puede estar el robot y las acciones que podría realizar.
- En este caso, los estados y acciones están predefinidos.

Actividad: Aprendizaje por refuerzo

- Se definen tres estados, en función de la mayor distancia detectada por el robot respecto a los obstáculos con sus sensores IR delanteros y tres acciones que el robot puede realizar:

Estado	Condición
S1	Mayor distancia de frente
S2	Mayor distancia a la derecha
S3	Mayor distancia a la izquierda

Acción	Comportamiento
A1	Avanzar recto
A2	Girar a la derecha
A3	Girar a la izquierda

- Una vez determinados los estados y acciones, hay que programar el algoritmo para que el robot sepa en qué estado se encuentra y cómo realizar cada una de las acciones.

Actividad: Aprendizaje por refuerzo

Parte 1: partiendo de la plantilla (plantilla_refuerzo_inicial.sb3), deberás completar 4 de las funciones: get_state, go_straight, turn_left, turn_right.

- get_state: obtiene el estado en el que se encuentra el robot. El resultado es un número entre 1 y 3, donde, 1 representa el estado S1, 2 el estado S2 y 3, el estado S3.

Estado	Condición	Comprobación
S1	Mayor distancia de frente	Front-C menor o igual que FrontRR y FrontLL
S2	Mayor distancia a la derecha	Front-RR menor o igual que FrontC y FrontLL
S3	Mayor distancia a la izquierda	Front-LL menor o igual que FrontC y FrontRR

Actividad: Aprendizaje por refuerzo

- go_straight: En este método, Robobo debería avanzar 1 segundo a velocidad 10.
- turn_right: giro a la derecha de 45º aproximadamente (1 segundo a velocidad -10,10).
- turn_left: giro a la izquierda de 45º aproximadamente (1 segundo a velocidad 10,-10).

Actividad: Aprendizaje por refuerzo

Parte 2: creación de la tabla Q ideal.

- La tabla se crea inicialmente con todos los valores a 0 porque, para el proceso de aprendizaje, el robot no conoce su entorno y tiene que aprender cual es la mejor acción para cada estado.
  - Al realizar la programación en Scratch, en lugar de tener una tabla, tenemos tres listas: state1, state2, state2, con tres elementos que representan las acciones 1, 2, y 3.

Actividad: Aprendizaje por refuerzo

- Inicialmente, las listas tendrán a 0 todos sus elementos:

- Antes de pasar a la parte de aprendizaje del algoritmo donde el robot aprenderá a navegar por el entorno utilizando estos tres estados y acciones, tendrás que crear una tabla Q ideal para obtener la mejor acción para cada estado. Hazlo en el bloque ideal_Qtable.

Actividad: Aprendizaje por refuerzo

- El algoritmo, seleccionará la mejor acción para cada estado basándose en el valor de estas listas. La mejor acción será la tenga un mayor valor en la lista.
- Para crear la tabla Q ideal pondremos un 1 en la mejor acción y un 0 en el resto de posibles acciones para cada estado.
- Recordamos lo que significa cada estado/acción:

Estado	Condición
S1	Mayor distancia de frente
S2	Mayor distancia a la derecha
S3	Mayor distancia a la izquierda

Acción	Comportamiento
A1	Avanzar recto
A2	Girar a la derecha
A3	Girar a la izquierda

Actividad: Aprendizaje por refuerzo

- Para probar lo realizado hasta ahora:

Se puede también utilizar un bucle “forever” en lugar de realizar un número fijo de iteraciones. En ese caso, el robot navegará por el entorno hasta que se detenga el programa.

Actividad: Aprendizaje por refuerzo

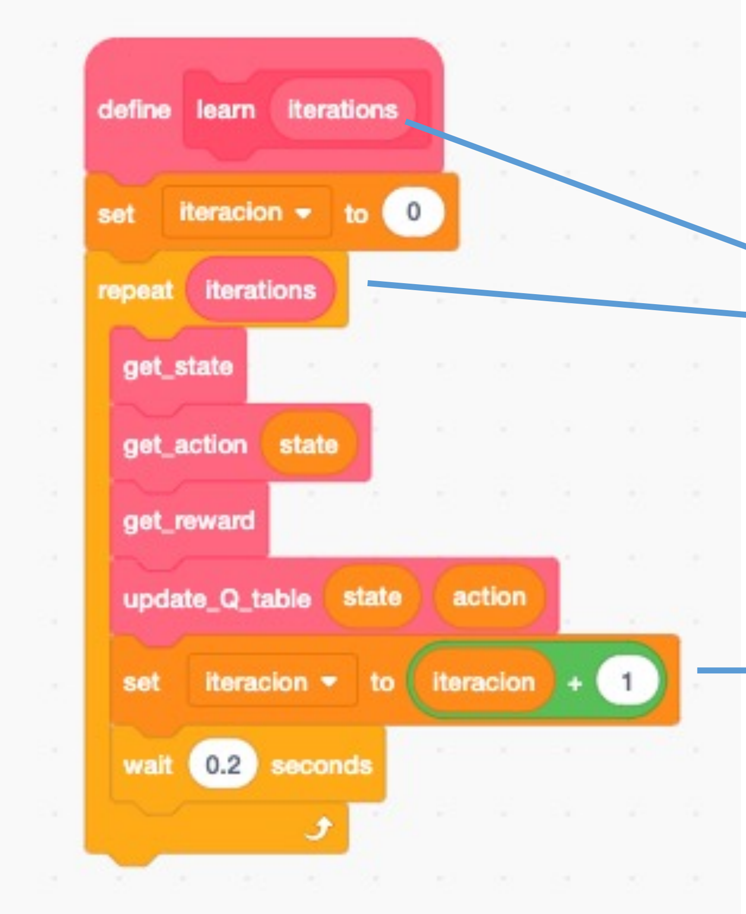
Parte 3: aprendizaje del algoritmo.

- A partir de la plantilla ”plantilla_refuerzo_aprendizaje.sb3”, rellenar la función “get_reward”.
- Calculará la recompensa recibida tras recibir la acción. Se recomienda comprobar la distancia a la que se encuentra el robot de las paredes con su IR delantero central (Front-C) y, establecer varios valores de recompensa en función de este valor.
  - Si el robot está demasiado cerca de la pared, su recompensa debería de ser muy inferior a la que obtendría si estuviese a una distancia media, de forma que pudiese continuar navegando sin dificultad.
  - Puedes probar inicialmente estableciendo una recompensa buena y una mala en función de la distancia y, a continuación, refinarlo estableciendo algún punto intermedio.

Actividad: Aprendizaje por refuerzo

- A continuación, el algoritmo actualizará la tabla Q con la función update_Q_table (función ya implementada en la plantilla).
- Para probar la parte de aprendizaje, rellenará el bloque "learn" que llevará a cabo todas las distintas etapas del aprendizaje:
  - Obtener estado del robot
  - Obtener acción y realizar la acción
  - Obtener recompensa tras la acción
  - Actualizar la tabla Q

Actividad: Aprendizaje por refuerzo



El aprendizaje se realiza durante un número de iteraciones establecido.

La variable iteración se usa solo para ver qué iteración está ejecutando el algoritmo.

Actividad: Aprendizaje por refuerzo

PROGRAMA PRINCIPAL

- Número de iteraciones del aprendizaje. Configurable.
- Tasa de aprendizaje. Configurable.
- Tasa de descuento. Configurable.
- Si está a 1 → fase aprendizaje. Influye en la manera de obtener la acción.
- Ratio de exploración. Configurable. Es un porcentaje, 4 representa el 40%.
- Inicialización de la tabla Q con ceros.
- Si está a 1 → fase de ejecución.

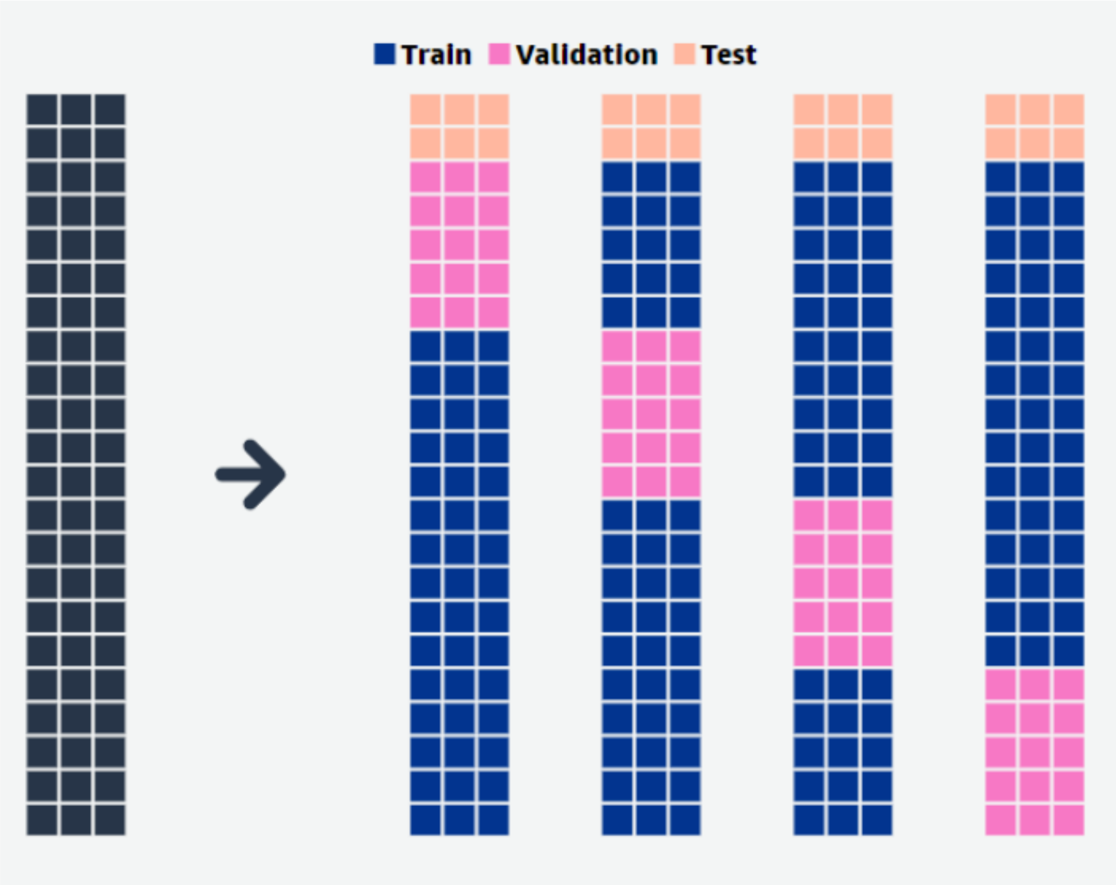
Actividad: entrenamiento de modelos

https://orangedatamining.com/download/

Actividad: entrenamiento de modelos

Partición del conjunto de datos

- Partición entrenamiento/test, no es la forma más utilizada, ya que presenta problemas.
- Se suele utilizar un método conocido como: validación cruzada.



https://mlu-explain.github.io/cross-validation/

Actividad: entrenamiento de modelos

Clasificación: evaluación

- Matriz de confusión
- Exactitud
- Sensibilidad
- Especificidad
- Precisión

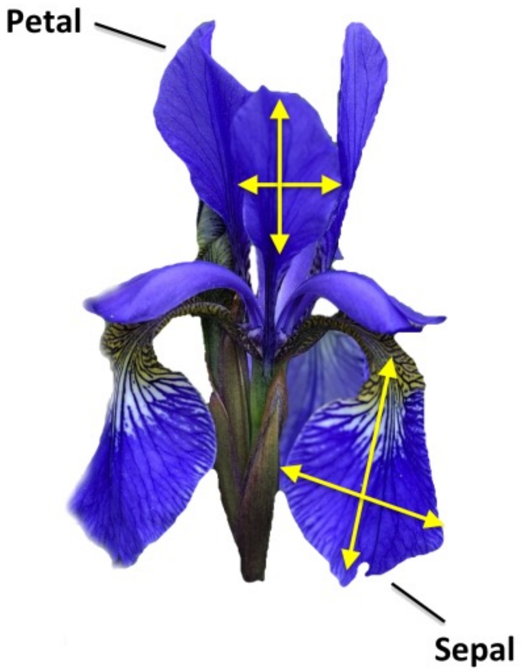
Matriz de confusión		Estimado por el modelo			
		Negativo (N)	*Positivo (P)*		
Real	*Negativo*	a: (TN)	b: (FP)		
	Positivo	c: (FN)	d: (TP)	**Precisión** ***("precision")*** Porcentaje predicciones positivas correctas:	d/(b+d)
		Sensibilidad, exhaustividad ("*Recall*") Porcentaje casos positivos detectados	**Especifidad (*Specifity*)** Porcentaje casos negativos detectados	**Exactitud ("accuracy")** Porcentaje de predicciones correctas *(No sirve en datasets poco equilibrados)*	
		d/(d+c)	a/(a+b)	(a+d)/(a+b+c+d)	

https://telefonicatech.com/blog/ml-a-tu-alcance-matriz-confusion

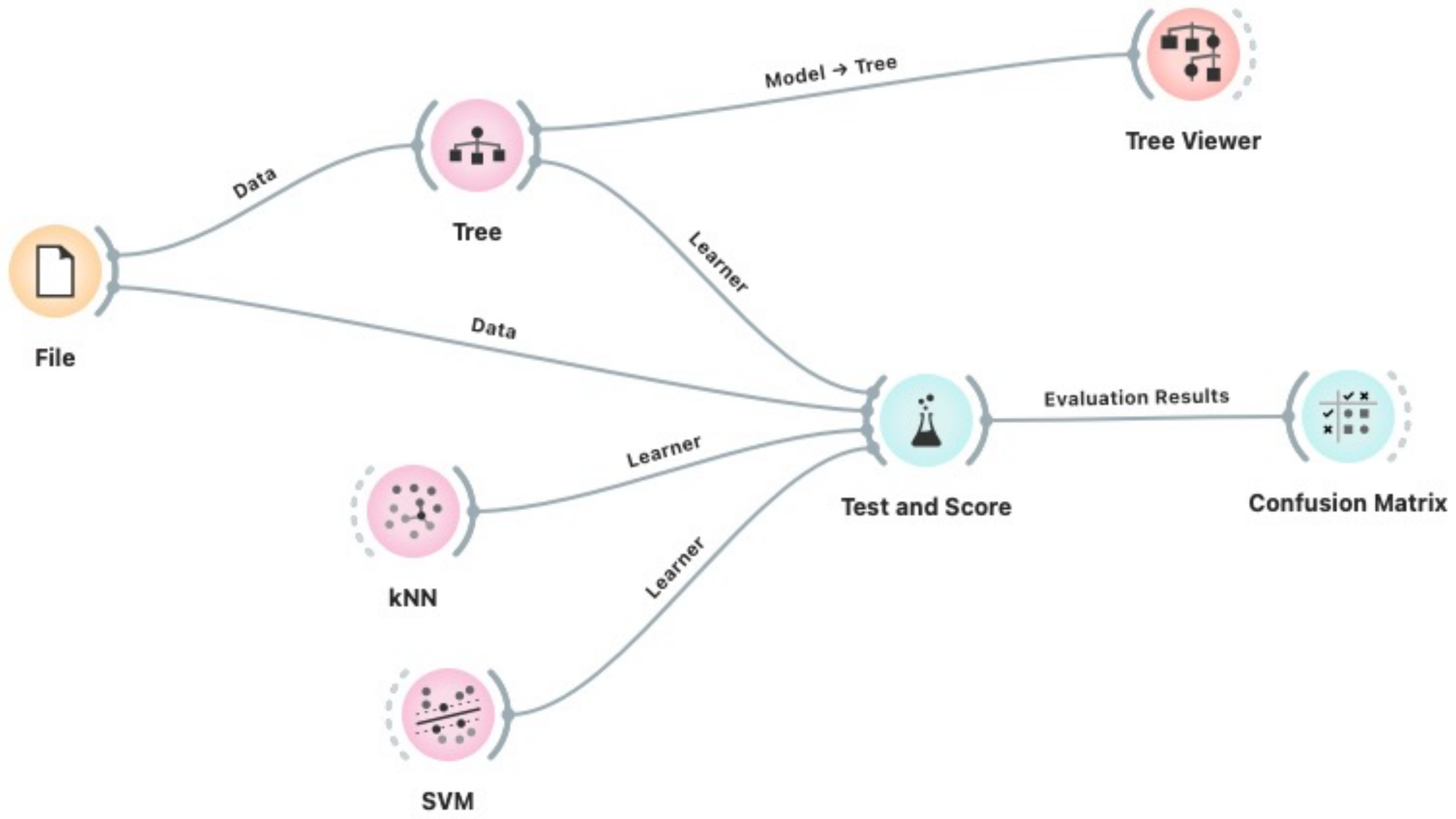
Actividad: entrenamiento de modelos

- Actividad 1: clasificación supervisada
  - Conjunto de datos: Iris
    - https://archive.ics.uci.edu/dataset/53/iris (ya incluido en Orange, no necesaria descarga)
    - Información del conjunto de datos:

Número de muestras	Valores perdidos	Atributos de entrada	Clases
150	No	4	3



Actividad: entrenamiento de modelos

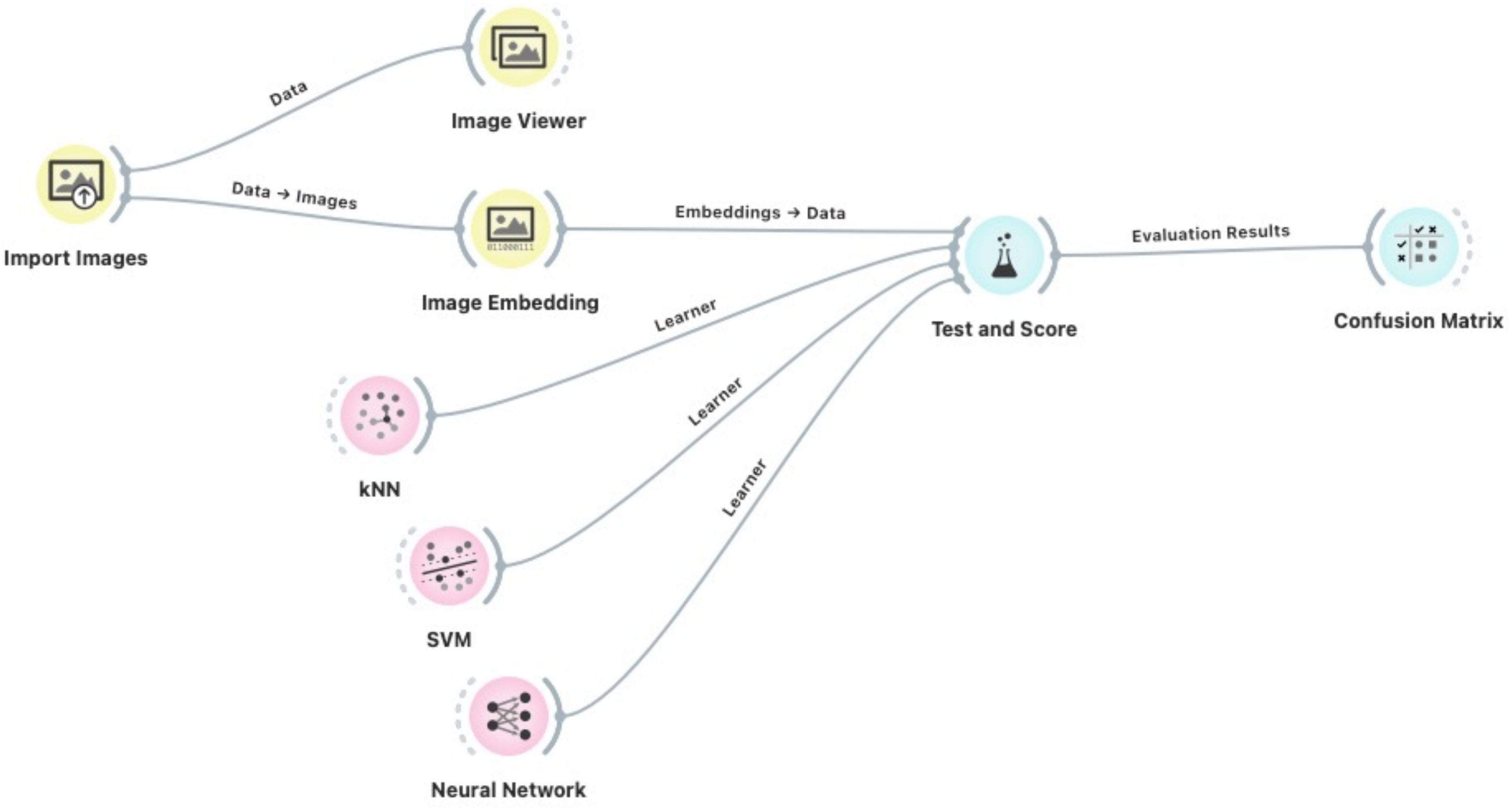


Actividad: entrenamiento de modelos

- Actividad 2: clasificación supervisada de imágenes
  - Utilizaremos un subconjunto de imágenes, extraídas de:
    - https://www.kaggle.com/datasets/marquis03/flower-classification
  - Las imágenes están disponibles para descargar en el aula virtual.

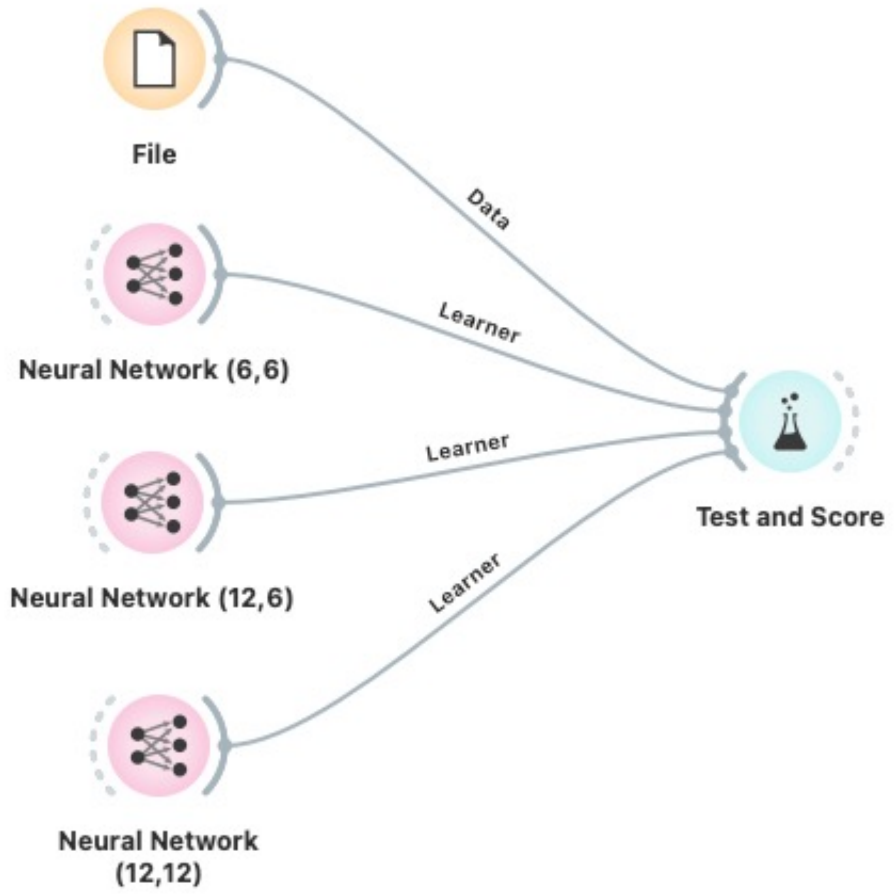



Actividad: entrenamiento de modelos

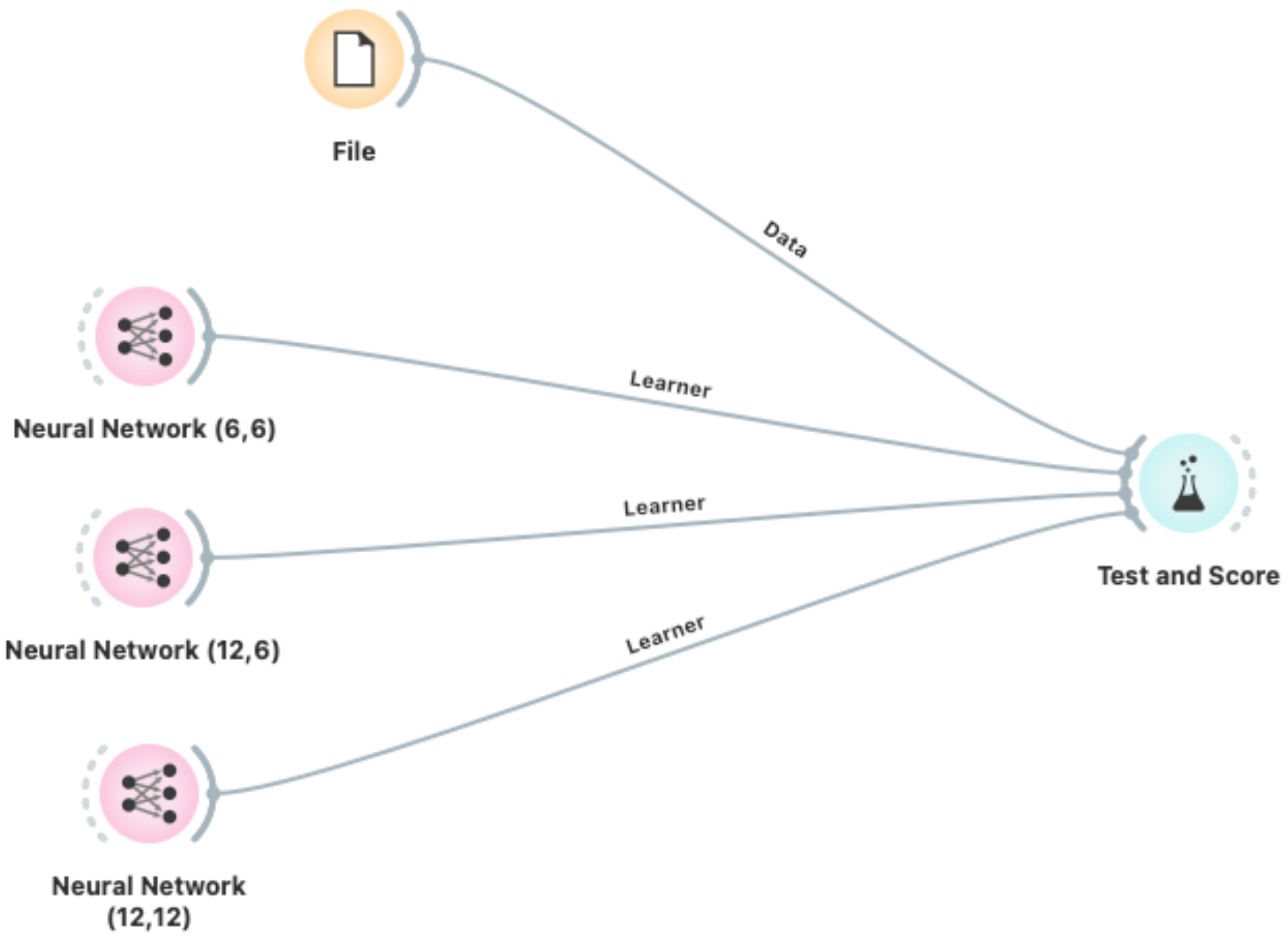


Actividad: entrenamiento de modelos

- Actividad 3: regresión
  - Utilizaremos el siguiente conjunto de datos:
    - https://www.kaggle.com/datasets/heitornunes/yacht-hydrodynamics-data-set
  - El conjunto de datos está disponible para descargar en el aula virtual



Actividad: entrenamiento de modelos



XUNTA DE GALICIA | CENTRO DE FORMACIÓN E RECURSOS DE FERROL

Inteligencia Artificial para la Sociedad

Alma Mallo

alma.mallo@udc.es

Diciembre de 2024