

Razonamiento Automático 2024/25 (GII)

Proyecto 2: Machine Learning

Fechas

- **Entrega:** Viernes, 17 de enero de 2025.
- **Defensa:** Jueves, 23 de enero de 2025 (lugar por especificar).

Proyecto Machine Learning (grupo, 2.00p)

1. Entrega en Moodle.
2. Se entregará `nombre_grupo-P2.zip` con:
 - Código fuente e instrucciones de compilación para Linux Manjaro.
 - PDF con:
 - Descripción de tecnologías implementadas y estado operativo.
 - Manual de uso (entrenamiento de bots y uso posterior).
 - Experimentos realizados y resultados.
 - Conclusiones obtenidas.
3. Criterios de evaluación:
 - [50%] **Funcionalidad:** funciona correctamente y sin fallos.
 - [35%] **Experimentos realizados:** pruebas con varios conjuntos de datos, análisis de resultados y conclusiones obtenidas sobre funcionamiento y rendimiento. Se valorará que se realicen nuevos experimentos motivados por las conclusiones de los anteriores.
 - [15%] **Calidad de la implementación:** código bien estructurado y documentado, orientado a datos y con adecuado nivel de generalidad y usabilidad.
4. Evaluación:
 - Se sumarán puntos por cada algoritmo incluido y por cada **modelo de aprendizaje**, hasta un máximo de 2 puntos.
 - Cada algoritmo o modelo será evaluado con los criterios de evaluación del punto 3.
 - Ver **tabla 1**: algoritmos, modelos y funcionalidades valorados.

Defensa (mixto, 1.00p)

1. Presentación, máximo de 8 minutos.
2. Debe incluir:
 - Algoritmos implementados: Funcionalidades y estado operativo.
 - Modelos de aprendizaje desarrollados.
 - Cómo se ha verificado que los modelos funcionan.
 - Experimentos realizados, resultados y conclusiones.

- Costes temporales del proyecto.
- Lecciones aprendidas.

BOTs ALE (individual, +1.25p)

1. Entrega en Moodle.
2. Se entregará `nombre_y_apellidos-ALE.zip` con:
 1. `readme.txt`: nombre completo, tecnologías utilizadas, arquitectura y otra información relevante.
 2. `howto.txt`: requerimientos e instrucciones de compilación.
 3. `makefile`: sistema de compilación.
 4. Subcarpeta `src/`: código fuente.
 5. Tras compilar, el ejecutable será `nombre_y_apellidos.bot`.
3. Se debe incluir todo lo necesario para compilar, especialmente librerías o fuentes.
4. Se utilizará la [MÁQUINA VIRTUAL DE CPC TELERA](#) como sistema objetivo. El bot debe funcionar ahí (*Linux Manjaro x86_64*).
5. Está **PROHIBIDO** leer/escribir ficheros durante la ejecución del bot.
6. El bot debe hacer uso **obligatoriamente** de la tecnología desarrollada por el grupo.

Campeonato

1. El juego será elegido en clase mediante votación por Moodle.
2. Dependiendo del juego, se realizará de 1 a 3 ejecuciones de cada bot. La puntuación obtenida será la media.
 - Si el juego lo permite, se harán enfrentamientos uno contra uno con sistema de torneo.
3. Los 10 mejores clasificados obtendrán puntuación extra según esta fórmula (E : nota extra, p : posición final),

$$E = 0.125(11 - p)$$

4. Los clasificados del 11 en adelante, cuyo bot funcione y no se quede parado, obtendrán todos +0.075 de nota.

Tabla 1: Algoritmos de ML por Nivel

Nivel	Nota	Algoritmo
0	0.50	Perceptrón
1	1.30	Red Neuronal con Backpropagation <ul style="list-style-type: none"> - +0.30 - Regularización - +0.25 - Validación - +0.15 - Selección de funciones de activación
2	2.00	Red Neuronal con pesos entrenados con Algoritmos Genéticos

Nivel	Nota	Algoritmo
3	3.00	Neuroevolución. Red Neuronal con Algoritmos Genéticos: evolución de arquitectura y pesos.
Adicional	+1.00	Sistema decisional (modelo de aprendizaje): <ul style="list-style-type: none"> - Varios algoritmos con decisión ponderada - Sistemas de voto, boosting, bagging o similares - Se valorará sistemas decisionales de diseño propio