Examen Aprendizaje Automático y Minería de Datos

Grado en Desarrollo de videojuegos, enero 2025

Instrucciones generales

El examen consta de dos partes, una teórica que vale 2 puntos y una práctica que vale 8 puntos.

La parte teórica se entregará en el campus con el nombre del alumno sin espacios y en pdf (Podéis usar word para exportar a PDF Archivo>exportar>crear pdf), word o txt.

La parte práctica se debe entregar en un Jupiter Notebook dentro de un zip o en su defecto en un fichero .py de Python normal. En el zip también debéis incluir vuestra librería con la implementación del perceptrón multicapa generalizado para múltiples capas y neuronas por capa. Esta librería se debe basar en el código entregado en la práctica 5 y 6 aunque puede tener modificaciones.

La práctica debe ser ejecutable sin intervención del profesor, si no ejecuta habrá penalización en la nota.

Se permite para la realización de la parte práctica y teórica el uso de los recursos del campus, apuntes y cualquier material de apoyo que traigáis. Para la parte teórica no tendréis acceso a internet, sólo al campus, pero para la parte práctica se os dará acceso a internet. **No está permitido el uso de ChatGPT o herramientas de IA similares ni establecer comunicación entre vosotros o con terceros.** Si se detecta, la práctica se considerará copiada y tendrá un 0. Tampoco está permitido el móvil durante la realización del examen.

Parte teórica (2 puntos)

Pregunta 1 (0.4 puntos)

Si disponemos de un dataset con ejemplos sin etiquetar, explica brevemente como podríamos afrontar el problema para poder construir un modelo de estos datos y que técnicas utilizarías.

Pregunta 2 (0.4 puntos)

Si disponemos de estas matrices de confusión donde las filas son los valores reales y las columnas los valores predichos de dos modelos entrenados con el mismo dataset. ¿Qué podemos decir de ambos modelos y de sus entrenamientos?

A	T	F
T	100	1
F	3	100

B	T	F
T	70	0
F	0	100

Pregunta 3 (0.4 puntos)

Si tenemos pocos datos de entrenamiento, ¿Qué estrategias podríamos utilizar para maximizar su utilidad? Explícala brevemente.

Pregunta 4 (0.4 puntos)

Estamos desarrollando un videojuego 2D mediante sprites pre-renderizados similar a juegos como Donkey Kong Country de Super Nintendo. No tenemos suficiente memoria para almacenar los sprites así que hemos decidido implementar un modelo de ML que nos permita reducir la memoria necesaria para cargar los sprites. Explica brevemente que tipo de modelo usuarias y cómo lo entrenarías.

Pregunta 5 (0.4 puntos)

Queremos crear un modelo que prediga el comportamiento de los precios del mercado de nuestro videojuego en base a juegos similares al nuestro. Para ello disponemos de un dataset con la descripción de miles de juegos y su evolución de precios durante su ciclo de vida. También disponemos de información acerca de su comportamiento en ventas. Con estos datos, ¿Qué modelo de machine learning crees que se adaptaría mejor para predecir el precio del juego? Se pueden combinar varios modelos si lo creéis necesario.

Tiempo estimado 30 mintos, tiempo máximo 45 minutos.

Parte práctica (8 puntos)

Tiempo estimado 2 h, tiempo máximo 2:30 h.

Disponemos del dataset **heart.csv** que se adjunta junto con el enunciado del examen. Dicho dataset contiene información acerca de pacientes con posibles problemas cardiovasculares. los campos del dataset son los siguientes:

- Age: edad del paciente (años)
- Sex: sexo del paciente (M: Hombre, F: Mujer)
- ChestPainType: tipo de dolor torácico (TA: Angina típica, ATA: Angina atípica, NAP: Dolor no anginoso, ASY: Asintomático).
- RestingBP: presión arterial en reposo [mm Hg].
- Colesterol: colesterol sérico (mm/dl).
- FastingBS: glucemia en ayunas (1: si FastingBS $>$ 120 mg/dl, 0: en caso contrario)
- RestingECG: resultados del electrocardiograma en reposo (Normal: Normal, ST: con anomalía de la onda ST-T (inversión de la onda T y/o elevación o

depresión del ST > 0,05 mV), HVI: con hipertrofia ventricular izquierda probable o definida según los criterios de Estes).
- FCmáx: frecuencia cardiaca máxima alcanzada (Valor numérico entre 60 y 202).
- ExerciseAngina: angina inducida por el ejercicio (Y: sí, N: no).
- Oldpeak: oldpeak = ST (Valor numérico medido en depresión).
- ST_Slope: la pendiente del segmento ST máximo del ejercicio (Up: pendiente ascendente, Flat: plano, Down: pendiente descendente).
- HeartDisease: clase de salida (1: cardiopatía, 0: normal).

Se desea contruir varios modelos para estudiar el comportamiento de los mismos para predecir un posible caso de cardiopatía con suficiente antelación.

Ejercicio 1 (0.5 puntos) Limpia el dataset y realiza las transformaciones necesarias para que los datos puedan ser aplicables a cualquier modelo de machine learning. Justifica en una celda de markdown las decisiones que has tomado.

Ejericcio 2 (0.5 puntos): Representa gráficamente los datos para mostrar la distribución de las clases.

Ejercicio 3 (2 puntos): Usa vuestro Perceptrón Multicapa de la práctica 5 y 6 para construir el modelo de predicción. Calcula accuracy y la matriz de confusión. Realiza una partición con random_state=0 y tamaño del test 25% El resultado mínimo que debéis conseguir de precisión es de un 84%.

El modelo puede tener cualquier capa, pero **debe existir una prueba realizada con más de una capa oculta**, aunque no sea el modelo definitivo que establezcáis. **Dejad bien claro cual es el modelo con el que finalmente os quedais.** en caso de que haya dos versiones.

Ejercicio 4 (1 puntos): Usa MLPClassifier de SKLearn para entrenar un modelo. Dicho modelo debe conseguir al menos un 84% de accuracy. La configuración puede ser diferente de la del ejercicio anterior

Ejercicio 5 (1 puntos): Usa el algoritmo KNN para conseguir una precisión del 84%.

Ejercicio 5 (1 puntos): Usa el algoritmo RandomForestClassifier para conseguir una precisión del 84%.

Ejercicio 5 (1 puntos): Dibuja las matrices de confusión y cualquier otra métrica que necesites para justificar con qué modelo te quedas. Justifica la respuesta.

Ejercicio 6 (1 puntos): implementa la opción de que la capa oculta pueda llevar una capa softmax en vez de una capa logistic. Para ello debes pasarle el parámetro output al constructor con el string “logistic” o “softmax” y en función de cual se elija, poder utilizar una u otra.