

Instrucciones:

- Esta es una actividad en grupos de no más de 3 integrantes.
 - Recuerden **unirse al grupo de canvas**
- No se permitirá ni se aceptará cualquier indicio de copia. De presentarse, se procederá según el reglamento correspondiente.
- Tendrán hasta el día indicado en Canvas.
 - No se confíen, aprovechen el tiempo en clase para entender todos los ejercicios y avanzar lo más posible.

Task 1 - Preguntas Teóricas

Responda a cada de las siguientes preguntas de forma clara y lo más completamente posible.

1. Explique la diferencia entre descenso de gradiente, descenso de gradiente por mini batches y descenso de gradiente estocástico. Asegúrese de mencionar las ventajas y desventajas de cada enfoque.
2. Compare y contraste técnicas de extracción de features (feature extraction) y selección de features (feature selection) en machine learning. De ejemplos de escenarios donde cada técnica sería más apropiada.
3. Describa la arquitectura y el funcionamiento de un perceptrón de una sola capa (un tipo de red neuronal sin backpropagation). Explique cómo aprende y la forma en la que actualiza sus parámetros.

Task 2 - Ejercicios Prácticos

Use Python en Jupyter Notebook para resolver los siguientes problemas relacionados a los temas que se vieron en clase. Asegúrese de citar todas las fuentes que utilice.

Task 2.1 - Gradiente Descendiente Estocástico

Implemente el descenso de gradiente estocástico, el descenso de gradiente y el descenso de gradiente por mini batches para una función polinómica de grado 3. Luego, grafique la función aproximada por cada uno de los métodos solicitados y la distribución real de puntos. Para esto considere que:

- El polinomio que debe usar es $2 * x^{**3} - 3 * x^{**2} + 5 * x + 3$
- Considere el uso de `np.polyval`
- Considere el uso de `np.polyder`
- Compare el tiempo de ejecución de cada uno de los métodos solicitados. De esto, mencione cuál método fue más rápido y por qué.
- Compare el fitness de cada método solicitado (recuerde hacer una evaluación numérica y no sólo visual). De esto mencione cuál método lo hizo mejor y por qué

Task 2.2 - Feature Selection

Tome de nuevo el dataset del laboratorio pasado referente a la clasificación de Partidas de League of Legends. Recuerde que lo puede descargar de este [enlace](#). Sobre este aplique **al menos 3 técnicas de feature selection** distintas, y con el dataset resultante vuelva a ajustar el modelo de Support Vector Machine (de la librería Sklearn). Recuerde que:

- Debe definir una métrica de desempeño y justificar el por qué de esta métrica
- Mida el performance de su modelo con las tres técnicas de feature selection y compare cada una de ellas. Es decir, debe tener al menos 4 valores a comparar (el obtenido en el laboratorio pasado, y al menos 3 de las técnicas de feature selection)

- De las medidas obtenidas, diga qué versión lo hizo mejor y por qué.
- Recuerde que para hacer parameter tuning, deben usar las mismas columnas seleccionadas después del feature selection
- Recuerde que su variable objetivo es "blueWins"

Task 2.3 - Perceptrón

Implemente un perceptrón de una sola capa para realizar una clasificación binaria del dataset de Iris (incluido en `sklearn.datasets.load_iris`). Visualice la frontera de decisión que fue aprendida por el perceptrón. Recuerde que:

- Debe cargar el dataset de Iris desde la librería mencionada
- Use solamente **dos** features, siendo estos *sepal length* y *sepal width*
- Asegúrese de que su clase *Perceptron* tenga los métodos de *fit* y *predict*
- Visualice la frontera de decisión
 - El eje Y debería ser *sepal width* y el eje X debería ser *sepal length*
- Debe definir una métrica de desempeño y justificar el por qué de esta métrica. Con dicha métrica, evalúe el rendimiento de su modelo. Es decir, diga si lo hizo bien o no y el por qué.

Para todos los subtasks de la parte 2 **evite el uso de herramientas de AI generativas (ChatGPT)**.

Nota: Para las partes donde se solicita que usen librerías, consideren usar [Sckit-learn](#). Por favor, **asegúrese de leer la documentación** de estas librerías para entender mejor los hiperparametros. Siempre recuerde, si tiene alguna duda o consulta, por favor comuníquese con el catedrático.

Entregas en Canvas

1. Jupyter Notebook respondiendo a cada task.
 - a. Incluyendo las preguntas del task 1
 - b. Dentro del Jupyter Notebook deben colocar el link a su repositorio en GitHub. Este puede permanecer como privado hasta la fecha de entrega.

Evaluación

1. [1 pt] Task 1
 - a. [0.2 pts] Pregunta 1
 - b. [0.4 pts] Pregunta 2 y 3
2. [4 pts.] Task 2
 - a. [1 pts] Subtask 2.1 y 2.3
 - b. [2 pts] Subtask 2.2