

Laboratorio Final  
Inteligencia Artificial Sección 01  
Primer Semestre 2025  
Ing. Max Cerna

**Temario Único**

26/05/2025

Nombre Completo: \_\_\_\_\_ Carné: \_\_\_\_\_

### INSTRUCCIONES GENERALES

- El laboratorio debe llevarse a cabo de forma individual
- Copias tendrán valor de 0

#### Serie Única (50 puntos):

Este laboratorio representa el 50% de la nota del examen final.

Una empresa del sector de Ciencia de Datos y Big Data desea optimizar su proceso de reclutamiento. Cuenta con datos demográficos, educativos y laborales de candidatos que han tomado cursos de capacitación con ellos. El objetivo es predecir si un candidato buscará cambiar de trabajo o si se quedará trabajando en la empresa.

Tu tarea consiste en construir y comparar modelos de clasificación utilizando tres enfoques distintos: **SVM (Máquinas de Vectores de Soporte)**, **Perceptrón (modelo lineal básico)** y una **Red Neuronal Multicapa (MLP)** empleando **Keras o TensorFlow**. Deberás interpretar sus métricas para evaluar cuál modelo tiene mejor desempeño.

- Usar el repositorio de Github Classroom: <https://classroom.github.com/a/reHpHMQo>
- Utiliza el dataset de [Kaggle](https://www.kaggle.com/arashnic/hr-analytics-job-change-of-data-scientists):
  - <https://www.kaggle.com/arashnic/hr-analytics-job-change-of-data-scientists>
- Preprocesamiento:
  - Realizar limpieza de datos
  - Tratamiento de valores faltantes
  - Codificación de variables categóricas.
- Divide los datos en conjuntos de entrenamiento y prueba (70% - 30%)
- Entrena un modelo con **SVM** utilizando `scikit-learn`.
- Entrena un modelo con **Perceptrón** (`sklearn.linear_model.Perceptron`).
- Entrena una **Red Neuronal** básica (al menos 1 capa oculta) usando **Keras o TensorFlow**.
- Calcula las siguientes métricas en el conjunto de prueba:
  - Accuracy
  - Precision

- Recall
- F1-score
- Muestra las métricas en una tabla comparativa
  - Archivo csv en `metrics/evaluation_report.csv`

modelo	accuracy	precision	recall	f1_score
svm	...	...	...	...
perceptron	...	...	...	...
neural_net	...	...	...	...

Se espera que, al ejecutar `python src/main.py`:

- Se prepare la data y se guarde en `processed/train.csv`
- Se entrene cada modelo.
- Se genere el archivo `metrics/evaluation_report.csv` con las métricas resultantes.

Estructura del repositorio:

```
|— data/
|   |— train.csv           # Dataset original que debes descargar
|— processed/
|   |— train.csv          # Dataset pre-procesado
|— metrics/
|   |— evaluation_report.csv # Informe de evaluación de modelos
|— src/
|   |— main.py            # Script principal
|— tests/                 # Pruebas unitarias
|   |— test_encoding.py
|   |— test_model_training.py
|   |— ...
|— requirements.txt       # Dependencias necesarias
|— README.md              # Documentación del proyecto
```

Criterio	Puntos
Ejecutar main.py sin errores y generar el archivo requerido	10
El preprocesamiento trata los valores faltantes correctamente	10
Codificación de variables categóricas está presente	10
Entrenar y evaluar correctamente el modelo SVM	10
Entrenar y evaluar correctamente el Perceptrón	10
Entrenar y evaluar correctamente una red neuronal en Keras	10
El archivo evaluation_report.csv contiene todas las métricas	10
Identificar cuál modelo tuvo mejor desempeño según f1_score y justificarlo	10
Comenta ventajas/desventajas observadas entre SVM, Perceptrón y Red Neuronal	10
Proponer al menos una mejora o justificación en los resultados obtenidos	10
Total	100 pts

Los últimos 3 puntos marcados con color deberán ser justificados y explicados en el archivo mis\_respuestas.txt en la raíz del proyecto

