

INTELIGENCIA ARTIFICIAL - ELP 8012 - UNIVERSIDAD DEL NORTE - EVALUACIÓN FINAL
PROGRAMA DE INGENIERÍA DE SISTEMAS
PROFESOR: EDUARDO ZUREK, PH.D.
Valor: La Evaluación Final vale 25% de la nota definitiva

Temas: Todos los temas considerados en la clase.

Objetivo general

Aplicar técnicas de aprendizaje automático supervisado y no supervisado para explorar, modelar y evaluar un conjunto de datos seleccionado por los estudiantes, con una **variable objetivo categórica**, y demostrar comprensión algorítmica mediante la implementación de un modelo en **lenguaje C**.

Condiciones del dataset

- Debe tener **al menos 500 observaciones** y **5 variables predictoras** de tipos variados (numéricas, categóricas, ordinales, textuales o temporales).
- La **variable objetivo** debe ser **categórica** (clasificación binaria o multiclase).
- Debe provenir de una fuente abierta (Kaggle, UCI, GitHub, etc.).
- Los estudiantes deben justificar su elección del dataset (relevancia y dominio de aplicación).

Preguntas / Tareas (25 ítems evaluables)

1. Comprensión de los datos

1. Describa el conjunto de datos seleccionado: fuente, dominio, tamaño, tipo de variables y problema a resolver.
2. Formule una **hipótesis de predicción** basada en la variable objetivo (¿qué quiere predecir y por qué?).
3. Realice una **exploración inicial (EDA)** identificando valores faltantes, outliers y distribución de variables.
4. Aplique **análisis de correlación o asociación** para seleccionar las variables más influyentes sobre la variable objetivo.
5. Genere **visualizaciones multivariadas** (gráficos de dispersión, boxplots, mapas de calor, etc.) que ayuden a entender los patrones de los datos.

2. Preprocesamiento

6. Realice el **tratamiento de datos faltantes**, codificación de variables categóricas y normalización/estandarización según sea necesario.
7. Divida el dataset en **conjunto de entrenamiento y prueba** (por ejemplo, 70/30).
8. Aplique una **reducción de dimensionalidad** (PCA o similar) y discuta los resultados obtenidos.

3. Aprendizaje no supervisado

9. Aplique un método de **clustering** (K-means, DBSCAN o jerárquico) sobre las variables predictoras.
10. Determine el **número óptimo de clusters** (por ejemplo, usando el método del codo o silueta).
11. Visualice los clusters en 2D o 3D y analice su relación con la variable objetivo original.
12. Aplique una técnica de **reducción de dimensionalidad no supervisada** (PCA, t-SNE o UMAP) para observar patrones o separaciones entre clases.

4. Aprendizaje supervisado

13. Entrene al menos **dos modelos de clasificación supervisada** (por ejemplo: Árbol de decisión, SVM, Regresión logística, KNN, Random Forest, etc.).
14. Compare el **rendimiento de los modelos** usando métricas apropiadas (exactitud, precisión, recall, F1-score, matriz de confusión).
15. Aplique **validación cruzada (k-fold)** y discuta la estabilidad de los modelos.
16. Use **ajuste de hiperparámetros (Grid Search o Random Search)** para optimizar uno de los modelos.
17. Interprete los **resultados de importancia de variables o pesos de los modelos** (feature importance o coeficientes).

5. Evaluación global e interpretación

18. Compare los resultados del aprendizaje supervisado con los del no supervisado. ¿Los clusters coinciden parcialmente con las clases reales?
19. **Mejora metodológica y optimización del modelo:**
Analice los resultados obtenidos y proponga e implemente **mejoras metodológicas concretas** para optimizar el desempeño del modelo o la calidad del aprendizaje. Estas mejoras pueden incluir:
 - Selección o **ingeniería de nuevas características (feature engineering)**.

- **Combinación o ensamblaje de modelos (ensemble learning)**, como bagging, boosting o stacking.
- **Uso de técnicas de balanceo de clases** (SMOTE, undersampling, oversampling) si existen clases desbalanceadas.
- **Reentrenamiento con diferentes configuraciones de parámetros** o con métodos de regularización.
- **Evaluación de nuevas métricas** que reflejen mejor el objetivo del problema (por ejemplo, AUC, balanced accuracy, Cohen's kappa).

Finalmente, justifique su propuesta explicando **cómo las modificaciones sugeridas podrían mejorar la capacidad predictiva, la interpretabilidad o la generalización del modelo.**

20. Concluya con una **discusión crítica**: ¿qué aprendió sobre el dataset, el modelo y la aplicabilidad del aprendizaje automático al problema real?

6. Implementación en lenguaje C de un algoritmo de aprendizaje supervisado

21. Selección y justificación del algoritmo:

Elija un algoritmo de aprendizaje supervisado que pueda implementarse desde cero en lenguaje C (por ejemplo, K-Nearest Neighbors, Regresión logística, Perceptrón, Naive Bayes, Árbol de decisión, etc.).

Explique brevemente por qué seleccionó ese algoritmo y en qué tipo de problemas es más efectivo.

22. Diseño de la estructura de datos y funciones:

Defina las estructuras de datos necesarias para representar las características, etiquetas y parámetros del modelo (por ejemplo, matrices, vectores, estructuras struct).

Describa las funciones principales del programa (lectura de datos, entrenamiento, predicción, cálculo de métricas, etc.) mediante un **diagrama o pseudocódigo**.

23. Entrenamiento y predicción:

Implemente en C el proceso de entrenamiento del modelo y la función de predicción sobre un subconjunto reducido del dataset (puede ser una versión simplificada del conjunto original).

Muestre fragmentos del código que evidencien el cálculo de los parámetros o la lógica de clasificación.

24. Evaluación del desempeño:

Evalúe el rendimiento del modelo implementado en C utilizando al menos una métrica de desempeño (por ejemplo, exactitud,

precisión o F1-score).

Compare brevemente los resultados obtenidos con los del modelo entrenado en Python u otra biblioteca de machine learning.

25. **Optimización y reflexión técnica:**

Analice las limitaciones de su implementación en C (eficiencia, escalabilidad, precisión numérica, facilidad de uso).

Proponga **mejoras o extensiones** posibles (por ejemplo, paralelización, modularización, manejo de archivos grandes, integración con librerías externas).

Productos entregables

- Entregables para el sábado 29 de noviembre antes de las 11:59 pm:
 - Informe técnico (notebook Jupyter) con código, gráficos y resultados.
 - Código fuente del algoritmo implementado en C, con documentación y comentarios.
 - Repositorio en GitHub con documentación reproducible.
- Presentación oral con diapositivas (aproximadamente 20 minutos) que se realizará el martes 2 de diciembre en el salón 31C de 7:30 am a 11:30 am. El profesor definirá el orden de los grupos, todos los estudiantes deben estar presentes todo el tiempo.

La solución se debe desarrollar en equipos de máximo 3 personas.

¡ÉXITOS!

Rúbrica de Evaluación — Proyecto Final de Inteligencia Artificial			
Sección	Criterio de evaluación	Descripción del desempeño esperado	Ponderación (%)
1. Comprensión de los datos	Descripción del dataset y planteamiento del problema	Identifica claramente la fuente, el dominio, la variable objetivo y el propósito del análisis. Demuestra comprensión del contexto y relevancia del problema.	5
	Exploración de datos (EDA)	Realiza análisis estadístico descriptivo, identifica outliers, valores faltantes y patrones significativos. Usa visualizaciones adecuadas.	5
Subtotal sección 1			10
2. Preprocesamiento de datos	Limpieza, codificación y escalado	Aplica correctamente técnicas de tratamiento de datos faltantes, codificación y normalización; justifica las decisiones tomadas.	5
	Reducción de dimensionalidad	Implementa PCA u otro método y analiza los resultados de manera interpretativa.	5
Subtotal sección 2			10
3. Aprendizaje no supervisado	Implementación de clustering	Aplica correctamente un método de clustering (K-means, DBSCAN o jerárquico), justifica la elección y discute resultados.	5
	Determinación y validación de clusters	Usa métricas o métodos (codo, silueta) para determinar el número óptimo de clusters y analiza su relación con las clases reales.	5
	Visualización e interpretación de patrones	Representa los clusters en espacios reducidos (PCA, t-SNE) y discute patrones emergentes.	5
Subtotal sección 3			15
4. Aprendizaje supervisado	Entrenamiento de modelos	Implementa y compara al menos dos modelos de clasificación supervisada. Justifica la elección de algoritmos.	5
	Evaluación y métricas	Usa métricas apropiadas (exactitud, F1, matriz de confusión). Interpreta correctamente los resultados.	5
	Validación y optimización	Aplica validación cruzada y ajuste de hiperparámetros; discute mejoras observadas.	5
	Interpretabilidad del modelo	Analiza la importancia de variables o coeficientes y relaciona con el problema original.	5
Subtotal sección 4			20
5. Evaluación global e interpretación	Integración de resultados supervisados y no supervisados	Analiza coherencias y diferencias entre ambos enfoques, aportando una interpretación crítica.	5

	Propuesta de mejora metodológica	Presenta mejoras sólidas (feature engineering, ensemble, balanceo, nuevas métricas) y justifica su impacto esperado.	5
	Discusión crítica final	Redacta una reflexión profunda sobre el proceso, los resultados y las limitaciones del modelo.	5
Subtotal sección 5			15
6. Implementación en C del algoritmo supervisado	Selección y justificación del algoritmo	Escoge un algoritmo adecuado, justifica su pertinencia y explica su funcionamiento teórico.	4
	Diseño de estructuras de datos y funciones	Desarrolla estructuras y funciones claras y eficientes; documenta el diseño con pseudocódigo o diagrama.	4
	Implementación funcional	Logra un programa en C que entrena y predice correctamente sobre un subconjunto del dataset.	4
	Evaluación y comparación de resultados	Calcula métricas básicas y compara el desempeño con el modelo equivalente en Python.	4
	Análisis técnico y optimización	Reflexiona sobre eficiencia, precisión y posibles mejoras (modularización, paralelismo, librerías, etc.).	4
Subtotal sección 6			20
7. Comunicación y presentación final	Claridad del informe y presentación oral	Entrega un informe técnico estructurado (Jupyter/PDF) con código reproducible y exposición oral clara, con dominio conceptual.	10
TOTAL GENERAL			100%