TAREA 1: Aprendizaje supervisado

Profesor: Fernando Crema Garcia
Fecha: 18 de Diciembre de 2025

La nota final será basada el promedio de cada uno de los modelos ponderados por un coeficiente **extra** dependiendo de la calidad del reporte asociado.

Regresión	R. Logística	Extra
R	RL	α

Fecha de entrega: 15/01/2026 23:59 pm Caracas.
Penalizaciones: 2 ptos / día. Máximo 10 ptos.
Nota final:
$$\text{mín}\left\{20\alpha\left(\frac{R+RL}{40}\right), 20\right\}$$

I. Introducción

I-A. Objetivos

El objetivo de esta actividad es reforzar las habilidades adquiridas hasta el momento para aprendizaje supervisado específicamente para los casos:

1. Regresión Lineal
2. Regresión con regularización (Lasso y Ridge)
3. Regresión Logística
4. Clasificación con k-vecinos.

Siempre asumiendo que, para cada caso, aplicamos correctamente los conocimientos de preprocesamiento, selección de modelos y pipeline de proyectos en aprendizaje supervisado.

II. Evaluación

II-A. Reproducibilidad

Cada una de las secciones tiene la misma ponderación y al final será la suma de todos los ejercicios que lograron resolver ponderada por un criterio del grupo docente (α).

Para lograr que el grupo docente reproduzca sus resultados, deben asignar al comienzo de su entregable la semilla referente a su **cédula de identidad**

II-B. Modelos entrenados

Sin importar la sección de la tarea que esté realizando, asuma que debe agregar un notebook **por sección** en el cual deben explicar toda la lógica de entrenamiento de los modelos desde preprocesamiento de datos, selección de variables, selección de hiper parámetros y evaluación de modelos. Se deja a juicio del estudiante agregar tanta información sea necesaria para justificar sus hallazgos (gráficos, tabla de resultados, etcétera) asumiendo que el Notebook va a ser corrido en un ambiente independiente [1].

De igual manera, se deben respetar las siguientes restricciones:

1. Un Jupyter Notebook con uso adecuado de los ítems de headings (#) separando cada nivel de acuerdo al tipo de modelo y nombre del mismo. Esta vez, se deja a juicio de ustedes la configuración del mismo.
2. Asuman que el Notebook lo va a leer una persona del equipo de toma de decisiones de una empresa pero que tiene conocimiento previo en Aprendizaje Automático. Sean tan detallados como sea posible justificando cada decisión. **No hay soluciones únicas**.
3. Incentivo la creatividad de ustedes y formará parte de la evaluación de α
4. **No se pueden usar soluciones que provee Kaggle para los casos de Regresión Logística y Clasificación**. Nunca es un problema revisar soluciones de terceros. Sin embargo, su solución debe ser producto de **su** trabajo. Plagios serán penalizados con nota mínima.
5. Incentivo la colaboración como descrito en el **código de honor** de la materia. Sin embargo, copias serán penalizadas con nota mínima.

email: fernando.cremagarcia@kuleuven.be

[1] Esto significa que no puede depender de archivos externos, si tienen alguna duda de cómo lograr esto contacte al grupo docente

III. Regresión

III-A. La Caja Fuerte de Totti

Un excéntrico magnate italiano (Francesco Totti) ha escondido la combinación de su caja fuerte (que contiene la receta *auténtica* de la pasta carbonara). Antes de morir, en vez de darnos la clave, solo dijo en su dialecto romano:

> *"Usa sempre er guanciale, nun coce l'ovo, niente ajo e niente cipolla, nun stai a fa er ragù, nè ojo, nè buro, nè strutto."*

Tu misión: descubrir la combinación y salvar la gastronomía romana. El magnate generó datasets de regresión aleatorios usando `make_regression` de sklearn, y escondió la clave en el **número de variables informativas**.

III-A1. Instrucciones:

1. Descarga tu dataset personalizado usando tu cédula de identidad desde `https://raw.githubusercontent.com/ucvia/ml-25-tarea01/main/datos/matrices/regression_data_{CEDULA}.csv`
2. El archivo contiene $p+1$ columnas y n filas que representan las características (predictores) y la respuesta por cada instancia del dataset.
3. Aplica el **camino de regularización Lasso** para identificar qué variables son realmente informativas.
4. Selecciona el mejor λ usando validación cruzada (puedes usar `LassoCV` para corroborar tus resultados pero debes reproducirlo con el resto de métodos de scikit-learn).
5. Identifica cuántas variables tienen coeficiente $\neq 0$ con el λ óptimo.
6. La combinación de la caja fuerte es la suma de:
   a) Número de muestras
   b) Número de variables totales
   c) Número de variables informativas

III-A2. Entregables: Dentro de su notebook, hará un informe claro y conciso donde expliquen y detallen como mínimo:

- Las dimensiones de x_i la i-ésima fila de X para cualquier i
- La dimensión de y_i la i-ésima respuesta de y para cualquier i
- La combinación (un número entero)
- Gráfico del Lasso Path mostrando coeficientes vs λ
- Justificación del λ seleccionado
- Explicación: ¿Por qué Lasso permite identificar variables informativas y Ridge no?

III-A3. Verificación de tu respuesta: Para comprobar que tu combinación es correcta en tu notebook:

```
abrir_caja_fuerte("TU_CEDULA", TU_COMBINACION)
```

Si la combinación es correcta, ¡obtendrás la receta secreta de la carbonara!

III-B. Parte 2: Comparación de Métodos de Regularización

Una vez descubierta la combinación secreta, el magnate quedó tan impresionado que te pide un último favor antes de entregarte la receta original escrita a mano por su *nonna*: demostrarle **por qué** Lasso fue la herramienta correcta para esta tarea.

III-B1. Objetivo: Comparar el rendimiento de tres métodos de regresión lineal sobre tu dataset:

- **OLS**: Mínimos cuadrados ordinarios (sin regularización)
- **Ridge**: Regularización L2 ($\|\beta\|_2^2$)
- **Lasso**: Regularización L1 ($\|\beta\|_1$)

III-B2. Instrucciones:

1. **Divide** tu dataset en entrenamiento (E %) y prueba (P %) siempre que $P+E=100$ usando `train_test_split` con `random_state=CEDULA`.
2. **Entrena** los tres modelos. Para Ridge y Lasso, usa validación cruzada para encontrar el mejor λ:
3. **Evalúa** cada modelo en el conjunto de prueba usando:
   - R^2 (coeficiente de determinación)
   - MSE (error cuadrático medio)
   - Número de coeficientes $\neq 0$
4. **Genera** una tabla comparativa con los resultados.
5. **Visualiza** los coeficientes de los tres modelos en un solo gráfico de barras agrupadas.

III-B3. Preguntas de Análisis: Responde las siguientes preguntas en tu informe:

1. Discuta sobre la elección de P y de E.
2. ¿Cuál modelo obtuvo mejor R^2 en el conjunto de prueba? ¿Era esperado?
3. ¿Por qué Ridge **no** puede reducir coeficientes exactamente a cero mientras que Lasso sí? *Pista: Investiga la geometría de las restricciones L1 vs L2.*
4. En un dataset con muchas variables ruidosas (como el tuyo), ¿cuál método es preferible para **interpretabilidad**? ¿Y para **predicción**?
5. ¿Qué sucede con OLS cuando $p > n$ (más variables que observaciones)? ¿Cómo resuelven esto Ridge y Lasso?
6. **Bonus:** Investiga y explica brevemente qué es **Elastic Net** y cuándo sería preferible sobre Lasso puro.

III-B4. Entregables de la Parte 2:

- Tabla comparativa de métricas (similar a la Tabla I)
- Gráfico de coeficientes de los tres modelos
- Respuestas a las 5 preguntas de análisis

Tabla I
EJEMPLO DE TABLA COMPARATIVA ESPERADA

Métrica	OLS	Ridge	Lasso
R^2 (test)	0.XX	0.XX	0.XX
MSE (test)	XX.XX	XX.XX	XX.XX
λ óptimo	–	X.XXX	X.XXX
Coefs $\neq 0$	XX	XX	XX

IV. Regresión Logística

Solo debe resolver uno de los siguientes ejercicios. Si desean resolver ambos, no hay ningún problema y ayudará en el α

IV-A. (Opción A) League of Legends

La popular empresa de desarrollo de videojuegos Riot Games recientemente introdujo un modelo que calcula la probabilidad de que un **equipo profesional** gane en base a situaciones similares de experiencias pasadas en su popular juego https://www.leagueoflegends.com/es-es. Una introducción sencilla al juego se puede ver en ¿Qué es League of Legends?

Citando su artículo sobre el tema dev: Probabilidad de Victoria con tecnología AWS: El porcentaje de probabilidad de victoria (PV) se obtiene a partir del número de equipos que se enfrentaron a una situación similar en el pasado y ganaron la partida. Nuestra estadística de PV se desarrolla utilizando un algoritmo de aprendizaje automático (AA) llamado xgboost, que tiene en cuenta muchos factores en su iteración actual.

IV-B. Variables usadas en este modelo

En su artículo, el equipo de datos de Riot explica cuáles son las variables usadas para su modelo:

1. Tiempo de la partida (tiempo dentro de la partida)
2. Porcentajes de oro (oro del jugador / oro total en la partida)
3. XP total de equipo
4. Número de jugadores con vida
5. Derribos de torretas
6. Asesinatos de dragón (si un equipo tiene un alma de dragón o no)
7. Baratija de Heraldo en el inventario
8. Temporizadores de inhibidor (cuánto tiempo tarda en reaparecer un inhibidor) para cada inhibidor
9. Temporizadores de Barón (tiempo hasta que expira la mejora de Barón del equipo)
10. Temporizador de Dragón Ancestral (tiempo hasta que expira la mejora del Dragón Ancestral del equipo)
11. Número de jugadores con el Barón activo
12. Número de jugadores con el Dragón Ancestral activo

Sin embargo, se han dado cuenta que el modelo funciona bien solo a nivel profesional y es complicado implementar una funcionalidad para su API de desarrollo por lo que XGBoost ha dejado de funcionar [2]

IV-C. Cómo se ve este modelo?

En la figura 1 podemos ver un ejemplo de la aplicación del modelo para explicar al usuario los cambios en probabilidad en el tiempo.

[2]XGBoost sigue funcionando... pero lo usaremos luego =D

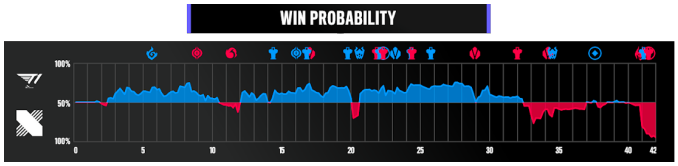

Figura 1. Obtenemos la probabilidad de victoria en el tiempo. Si la probabilidad de que el equipo azul es mayor que la del rojo, la gráfica tiene color azul y, en caso contrario, toma color rojo. En la parte superior de la gráfica hay íconos que se refieren a eventos de importancia que modifican radicalmente la probabilidad de victoria. Por ejemplo, asesinatos entre campeones, torres tiradas o dragones conseguidos.

IV-D. Necesitamos tu ayuda

El equipo de datos de Riot nos ha pedido realizar un modelo cuyo output sea: **Probabilidad de que el equipo Azul Gane luego de 10 minutos**. Para hacer el problema más sencillo, usamos un conjunto de datos donde en el minuto 10 almacenamos variables similares a las que usa Riot.

Nuestro trabajo, además de conseguir el modelo, es explicar detalladamente cuáles variables seleccionamos y una breve explicación de la importancia en el modelo. Para ello, nos han pedido revisar la sección de Selección de variables y probar **dos** métodos que nos llamen la atención.

Además de ello, piden como entrega un Pipeline que involucre también el método selección de modelos justificado en su entrega.

Como tienen poco tiempo para revisar el informe, piden con ahínco que los resultados sean explicados con gráficas que puedan mostrar al equipo de negocios y decidir si seguir adelante con el proyecto (usando XGBoost) y datos nuevos [3].

IV-E. Los datos disponibles

Usaremos el dataset disponible en Kaggle Diamond ranked games after 10 minutes. Los datos están divididos en 3 secciones:

1. gameId
2. Variables del equipo azul con prefijo blue.
3. Variables del equipo rojo con prefijo red.

Para cada tipo red y blue tenemos las variables:

1. **Wins**: Ganó el equipo o no.
2. **WardsPlaced**: Número de centinelas puestos por el equipo.
3. **WardsDestroyed**: Número de centinelas destruídos por el equipo.
4. **FirstBlood**: Tuvieron primer asesinato o no.
5. **Kills**: Número de asesinatos.
6. **Deaths**: Número de muertes.
7. **Assists**: Número de asistencias.
8. **EliteMonsters**: Número de monstruos élite asesinados.
9. **Dragons**: Número de dragones asesinados.
10. **Heralds**: Número de heraldos asesinados.
11. **TowersDestroyed**: Torres destruídas.
12. **TotalGold**: Oro total.

[3]Uno de los proyectos puede ser hacer esto!

13. **AvgLevel**: Average del nivel de las leyendas.
14. **TotalExperience**: Experiencia total.
15. **TotalMinionsKilled**: Número total de minions asesinados.
16. **TotalJungleMinionsKilled**: Número total de minions de la jungla asesinados.
17. **GoldDiff**: Diferencia de oro.
18. **ExperienceDiff**: Diferencia de experiencia.
19. **CSPerMin**: Número de minions asesinados por minuto.
20. **GoldPerMin**: Oro por minuto.

Una sección de preprocesamiento de datos es necesaria y pueden usar las gráficas que deseen siempre que expliquen detalladamente lo que hacen.

IV-F. (Opción B) Cáncer de mama

El hospital universitario de Roma (Policlínico) desea iniciar un estudio sobre el cáncer de mama en mujeres. Para ello, te pide como evaluación de prueba que uses el conjunto de datos disponibles en Scikit llamado Breast Cancer. Tienen un problema importante: no entienden bien los datos. Además de conseguir un modelo que calcule la probabilidad de que una mujer tenga cáncer de mama debes explicar detalladamente cómo funciona tu modelo. Para ello, recomendaron el uso de Curvas ROC, F1-score y AUC. Pero, sinceramente, no entienden muy bien cada uno de esos términos. Es labor nuestra explicarles lo que pueden hacer!

IV-F1. Datos del Conjunto Breast Cancer Wisconsin: Link original del dataset disponible en UCI Breast Cancer

El conjunto de datos contiene las siguientes características:

- **Muestras totales:** 569
- **Clases:** Tumores malignos (M) y benignos (B)
- **Atributos:** 30 atributos numéricos relacionados con las propiedades de los núcleos celulares como:
  - Radio medio
  - Textura media
  - Perímetro medi

IV-F2. Cómo usar los datos?: Existen diferentes maneras recomendamos usar Sklearn datasets load breast cancer . Sin embargo, existen alternativas como

```
pip install ucimlrepo

from ucimlrepo import fetch_ucirepo

# fetch dataset
breast_cancer_wisconsin_diagnostic = fetch_ucirepo(id=17)

# data (as pandas dataframes)
X = breast_cancer_wisconsin_diagnostic.data.features
y = breast_cancer_wisconsin_diagnostic.data.targets

# metadata
print(breast_cancer_wisconsin_diagnostic.metadata)

# variable information
print(breast_cancer_wisconsin_diagnostic.varial
```

IV-F3. Qué busca el equipo médico?:

1. **(Muestreo)** Si tuviesen que seleccionar k mujeres de un universo de N de manera que la muestra sea representativa, ¿qué técnica utilizarían? Justifique.
2. **(Preprocesamiento)** ¿Es necesario estandarizar las variables antes de entrenar el modelo? ¿Por qué?
3. **(Interpretación)** Interprete los 5 coeficientes más importantes del modelo. ¿Qué significa que un coeficiente sea positivo o negativo en términos médicos? Calcule los *odds ratios* correspondientes.
4. **(Desbalance)** ¿Cómo afecta el desbalance de clases al modelo? Proponga al menos dos estrategias para mitigarlo.
5. **(Umbral)** ¿Qué umbral de probabilidad recomendaría para clasificar? Considere que en contexto médico, ¿es peor un falso negativo o un falso positivo?
6. **(Métricas)** Investigue:[4] ¿Por qué usar AUC y F1-score además de la exactitud? Ilustre con un ejemplo dónde la exactitud sea engañosa.

[4]Esta sección es opcional