Práctica Calificada 2

Machine Learning para CCSS y Gestión Pública

10 de noviembre de 2025

Instrucciones Generales

- **Grupos:** El trabajo será presentado por grupos de 4 integrantes.
- **Entregable:** Un único Jupyter Notebook (.ipynb), que contenga todo el código, los resultados (gráficos y salidas de celdas) y las respuestas escritas a las preguntas de interpretación.
- **Puntaje Total:** 20 puntos.
- **Fecha límite:** La fecha límite para la entrega es el lunes 17 de noviembre hasta las 11:59 pm. Se debe subir su solución a la actividad correspondiente creada en la plataforma de Paideia.
- **Cualquier duda:** Por favor, no duden en escribirme por correo a zdelacruzl@pucp.edu.pe.

Regularización y Métodos de Ensamble (20 puntos)

En este ejercicio, aplicarán y compararán diferentes modelos para predecir variables de interés en dos datasets. Todas las preguntas de interpretación deben ser respondidas directamente en el notebook, debajo de las celdas de código correspondientes.

0.1. Parte 1: Regularización para Predecir Precios de Viviendas (10 puntos)

Utilizarán el dataset `Boston Housing`, que contiene información sobre viviendas en los suburbios de Boston. El objetivo es predecir el valor de las viviendas ocupadas por sus propietarios (`medv`).

URL del dataset: https://raw.githubusercontent.com/qlabpucp/datasets/main/datasets/boston.csv

0.1.1. Preparación de Datos (3 puntos)

a) Cargue el dataset desde la URL en un DataFrame de Pandas.

b) Explore la base de datos, verifique si hay valores nulos y revise los tipos de datos. No hay variables categóricas que necesiten ser convertidas.

c) Separe los datos en predictores (`X`) y la variable objetivo (`y`, la columna `medv`).

d) Divida los datos en un conjunto de entrenamiento (80 %) y un conjunto de prueba (20 %). Use un `random_state=42`.

e) Estandarice las características en `X` usando `StandardScaler`. Ajuste el escalador **solo** con los datos de entrenamiento y luego transforme ambos conjuntos.

0.1.2. Implementación e Interpretación de Lasso y Ridge (7 puntos)

a) **Lasso:** Entrene un objeto `LassoCV` con 10 pliegues (`cv=10`) sobre los datos de entrenamiento para encontrar el `lambda` óptimo. Imprima el valor encontrado y grafique.

b) Muestre los coeficientes del mejor modelo Lasso. En este dataset, ¿Lasso eliminó alguna variable (estableció su coeficiente en cero)? Basado en los coeficientes no nulos, ¿cuáles parecen ser los predictores más importantes según el modelo?

c) **Ridge:** De manera similar, entrene un objeto `RidgeCV` con 10 pliegues para encontrar el `lambda` óptimo. Imprima el valor encontrado y grafique.

d) Muestre los coeficientes del mejor modelo Ridge. Compare las magnitudes de los coeficientes de Ridge con los de Lasso. ¿Ridge establece algún coeficiente exactamente en cero? Explique por qué esta es una diferencia entre ambos métodos de regularización (L_1 vs. L_2).

e) **Evaluación Final:** Evalúe el rendimiento de ambos modelos finales (Lasso óptimo y Ridge óptimo) sobre el **conjunto de prueba**. Calcule y reporte el Error Cuadrático Medio (MSE) para cada uno. ¿Qué modelo tuvo un mejor rendimiento predictivo en este caso?

0.2. Parte 2: Métodos de Ensamble para Predecir Ventas y Comparación Final (10 puntos)

Ahora, utilizarán el dataset `Carseats` para predecir ventas de sillas para autos infantiles.

URL del dataset: https://raw.githubusercontent.com/qlabpucp/datasets/main/datasets/carseats.csv

0.2.1. Preparación de Datos (2 puntos)

a) Cargue el dataset. Elimine la primera columna si es un índice innecesario.

b) Convierta las variables categóricas a variables dummy.

c) Separe los datos en predictores (`X`) y la variable objetivo (`y`, la columna `Sales`).

d) Divida los datos en un conjunto de entrenamiento (80 %) y un conjunto de prueba (20 %), usando `random_state=10`.

0.2.2. Modelos de Ensamble: Versión Base vs. Optimizada (5 puntos)

a) **Modelos Base a Criterio Propio:**

- Entrene un `BaggingRegressor` usando hiperparámetros que considere razonables.
- Entrene un `RandomForestRegressor` usando hiperparámetros que considere razonables.
- Entrene un `GradientBoostingRegressor` usando hiperparámetros que considere razonables.

Para cada modelo, evalúe su rendimiento calculando el MSE sobre el **conjunto de prueba**.

b) **Modelos Optimizados con GridSearchCV:**

- Optimice un `BaggingRegressor` usando `GridSearchCV` (`cv=5`). Pruebe con: {`'n_estimators': [50, 100, 200]`}.
- Optimice un `RandomForestRegressor`. Pruebe: {`'n_estimators': [100, 200], 'max_features': ['sqrt', 5, 10]`}.Indique cuales son las variables mas importantes segun el modelo.
- Optimice un `GradientBoostingRegressor`. Pruebe: {`'n_estimators': [100, 200], 'learning_rate': [0.01, 0.1], 'max_depth': [3, 5]`}.

Para cada modelo, evalúe el MSE del **mejor estimador** encontrado sobre el conjunto de prueba.

c) **Pregunta de Interpretación (Impacto de la Optimización):** Compare el MSE de los modelos base con el de los modelos optimizados. ¿Mejoró el rendimiento después de usar `GridSearchCV`? ¿Por qué es importante el ajuste de hiperparámetros?

0.2.3. Comparación Final y Conclusión (3 puntos)

a) Cree una tabla resumen que compare el MSE en el **conjunto de prueba** para los 3 modelos optimizados ajustados con `GridSearchCV`. ¿Qué modelo tiene el mejor rendimiento?

b) Basado en la tabla de comparación final, ¿qué modelo le recomendaría a la empresa para predecir las ventas? Justifique su elección considerando el **rendimiento predictivo** (MSE) y la **interpretabilidad** de los modelos.