
Universidad de
SanAndrés

# I302 - Aprendizaje Automático y Aprendizaje Profundo

**2do Semestre 2024**

Trabajo Práctico 2

---

**Fecha de entrega:** Domingo 1 de septiembre, 23:59 hs.

**Formato de entrega:** Los archivos desarrollados deben ser entregados en un archivo comprimido .zip a través del Campus Virtual, utilizando el siguiente formato de nombre de archivo: *Apellido_ Nombre_ TP2.zip*. Se aceptará únicamente 1 archivo por estudiante. En caso de que el nombre del archivo no cumpla con la nomenclatura especificada, el trabajo no será corregido.

Dentro del archivo .zip, debe incluirse un Jupyter Notebook llamado *Entrega_ TP2.ipynb* con las respuestas a los problemas y los gráficos resultantes. Puede agregar resultados o análisis adicionales si lo considera necesario. Se recomienda fuertemente no realizar todo el desarrollo dentro del Jupyter Notebook; en su lugar, se sugiere usar archivos .py para desarrollar el código, siguiendo las buenas prácticas de programación y modularización vistas en clase.
Se recomienda seguir la estructura sugerida al final del trabajo práctico.

---

# Trabajo Práctico 2: Regresión

El objetivo de este trabajo es desarrollar y evaluar diversos modelos de regresión para estimar el precio de venta de una SUV (Sport Utility Vehicle) de Toyota. El dataset *Precios de SUVs* contiene información sobre los precios en dólares de tres tipos de vehículos SUV de la marca Toyota, publicados en la página de Mercado Libre Argentina. Además, incluye características adicionales relacionadas con cada publicación. Previamente, el conjunto de datos fue dividido en subconjuntos de desarrollo (*toyota_ dev.csv*) y prueba (*toyota_ test.csv*).

**NOTA**: Use exclusivamente NumPy para la implementación de funciones y/o clases; no se permite el uso de librerías de Machine Learning como scikit-learn.

1. El primer paso consiste en crear una serie de gráficos para explorar las relaciones entre diferentes variables del dataset de desarrollo. Para ello, podrán hacer uso de las librerías Matplotlib o Seaborn.

   *a*) Realizar un análisis de la cobertura en las características para entender cómo se distribuyen los datos e identificar posibles sesgos o patrones en las variables de entrada.

      1) Generar histogramas para visualizar la distribución de tipos de vehículo, años de fabricación y kilómetros recorridos.
      2) Crear gráficos de dispersión para analizar la cobertura de datos en el espacio de kilómetros recorridos y años de fabricación, diferenciando por tipo de vehículo.
      3) Visualizar la distribución de la variable "Motor" según el tipo de vehículo. ¿Cómo podría agrupar o simplificar las categorías para obtener un análisis más claro?
      4) Comparar la distribución de los tipos de transmisión en función del tipo de vehículo utilizando gráficos adecuados.
      5) Analizar cómo se distribuye la variable "Color" entre los distintos tipos de vehículo.

   *b*) Realizar un análisis de correlación entre las características y el precio de venta. Este análisis puede guiar la selección de variables en modelos predictivos.

      1) Visualizar cómo varía el precio de venta en función de los kilómetros recorridos por el vehículo. ¿Observa la misma tendencia para los diferentes tipos de vehículo?
      2) Comparar la distribución de precios de venta entre los distintos tipos de SUV. ¿Identifica la presencia de outliers?
      3) Analizar la distribución del precio de venta según el año del vehículo.
      4) Evaluar la utilidad de la variable "Color". ¿Existen diferencias significativas en el precio de venta según el color del vehículo? ¿Hay algún modelo cuyo precio parezca estar influenciado por el color?

5) Comparar cómo varía el precio de venta según el tipo de combustible utilizado. ¿Existe algún tipo de vehículo cuyo precio de venta esté condicionado por el tipo de combustible?
6) Comparar los precios de venta en función del tipo de transmisión del vehículo.
7) Graficar la relación entre el tipo de vendedor y el precio de venta. ¿Hay alguna tendencia o patrón en los precios en función del tipo de vendedor?
8) Generar la matriz de correlación entre las características y el precio de venta. ¿Qué puede deducir a partir de estos resultados?

*c*) OPCIONAL: Agregar una o más visualizaciones que considere útiles para generar un insight valioso y mejorar la comprensión del dataset.

2. Dividir el conjunto de desarrollo en dos subconjuntos: uno de entrenamiento (80 %) y otro de validación (20 %). Luego, implementar los siguientes modelos, ajustando los hiperparámetros de cada uno, y reportar los valores de $\text{RMSE}_{\text{val}}$, $\text{MAE}_{\text{val}}$ y $\text{R}^2{}_{\text{val}}$ sobre el conjunto de validación:

   *a*) Regresión lineal con regularización L2 (`RidgeRegression`). Pueden optar por utilizar regresores no lineales, definidos por ustedes según consideren adecuado.

   *b*) Regresión localmente ponderada (`LocallyWeightedRegression`), utilizando como función de ponderación

   $$K(x, x_i) = \exp\left(-\frac{||x - x_i||^2}{2\tau^2}\right),$$

   donde $x$ es el punto de predicción, $x_i$ son los puntos de datos de entrenamiento, y $\tau$ es el parámetro de ancho de banda.

   *c*) Regresión no lineal (`NonLinearRegression`) empleando regresores con parámetros no lineales que deberán seleccionar o desarrollar según su criterio.

   **NOTA:** Es importante tener en cuenta que el objetivo es modelar el precio de venta de SUVs, por lo que deberán elegir un enfoque que capture de manera efectiva la relación no lineal entre las variables.

   **NOTA:** Para entrenar los modelos, pueden utilizar cualquier método de optimización que consideren adecuado. Por ejemplo, pueden emplear `scipy.optimize`, gradiente descendiente o el método de Newton.

3. Evaluar el rendimiento de los modelos `RidgeRegression`, `NonLinearRegression` y `LocallyWeightedRegression` de la siguiente manera:

   *a*) Utilizando la metodología de validación cruzada (CV) con 5 folds sobre el conjunto de desarrollo. Reportar los valores de $\text{RMSE}_{\text{CV}}$, $\text{MAE}_{\text{CV}}$ y $\text{R}^2{}_{\text{CV}}$ obtenidos.

   *b*) Evaluando el rendimiento de los modelos sobre el conjunto de prueba para obtener una estimación del rendimiento general en datos no vistos. Reportar los valores de $\text{RMSE}_{\text{test}}$, $\text{MAE}_{\text{test}}$ y $\text{R}^2{}_{\text{test}}$ obtenidos.

*c*) Comparar los resultados obtenidos en los dos incisos anteriores con los del split de entrenamiento-validación realizado previamente.

1) Elaborar una tabla comparativa que muestre los valores de RMSE, MAE y $R^2$ para cada modelo en los siguientes conjuntos:
   - Conjunto de validación del split de train-val: $\text{RMSE}_{\text{val}}$, $\text{MAE}_{\text{val}}$ y $\text{R}^2{}_{\text{val}}$.
   - Validación cruzada: $\text{RMSE}_{\text{CV}}$, $\text{MAE}_{\text{CV}}$ y $\text{R}^2{}_{\text{CV}}$.
   - Conjunto de prueba: $\text{RMSE}_{\text{test}}$, $\text{MAE}_{\text{test}}$ y $\text{R}^2{}_{\text{test}}$.

   Realizar un breve análisis de los resultados, explicando las variaciones de las métricas entre los diferentes conjuntos. Comparar la performance de los distintos modelos y discutir las posibles razones detrás de las diferencias observadas.
2) Para cada conjunto (train, validation, "held out" de CV y test), graficar el histograma de los residuos absolutos de los tres modelos implementados. Seleccionar un número de bins adecuado que capture la tendencia general de los residuos, evitando una representación demasiado errática.

   **NOTA**: Una alternativa al histograma es utilizar un gráfico de estimación de densidad de kernel (KDE, Kernel Density Estimation). El KDE permite visualizar las distribuciones de los residuos de los tres modelos en un mismo gráfico de manera más clara, especialmente cuando los histogramas se vuelven difíciles de interpretar debido a la superposición de barras.
3) Para cada conjunto (train, validation, "held out" de CV y test) y para cada uno de los tres modelos implementados, graficar el diagrama de dispersión de las predicciones $\hat{y}$ frente a los valores verdaderos $y$. Organizar los gráficos en una grilla de 4 filas por 3 columnas, donde cada fila corresponda a un conjunto y cada columna a un modelo.
4) Para cada uno de los tres modelos implementados, graficar la variación del precio predicho en función del año. Las predicciones deberán realizarse manteniendo todas las demás features fijas en sus valores promedio o cero, de modo que se observe únicamente el efecto del año en el precio según cada modelo.

**Estructura Sugerida para la Entrega del Trabajo Práctico**

Para organizar el desarrollo de este trabajo práctico de manera efectiva, recomendamos modularizar las diferentes funcionalidades en archivos .py y carpetas separadas de forma de facilitar la reutilización de código y la depuración. Una posible estructura de entrega podría ser:

```
Apellido_Nombre_TP2.zip
|- data/                  # Carpeta para los datos del proyecto
    |- raw/                   # Datos originales sin modificar
        |- toyota_dev.csv
        |- toyota_test.csv
    |- processed/             # Datos procesados y curados

|- src/                   # Carpeta para el código fuente del proyecto
    |- utils.py               # Funciones auxiliares
        |- save_results()          # Para guardar resultados
        |- load_model()            # Para cargar un modelo guardado

    |- metrics.py             # Funciones para calcular métricas
        |- rmse()
        |- mae()
        |- r2()

    |- preprocessing.py       # Funciones para el preprocesamiento
        |- one_hot_encoder()
        |- normalize()
        |- handle_missing_values()

    |- models.py              # Clases para los modelos de ML
        |- class RidgeRegression()
        |- class NonLinearRegression()
        |- class LocallyWeightedRegression()

    |- data_splitting.py      # Funciones para dividir los datos
        |- train_val_split()
        |- cross_val()

|- notebooks/             # Carpeta para Jupyter Notebooks
    |- Entrega_TP2.ipynb    # Respuestas de todos los ejs del TP

|- requirements.txt         # Especificar dependencias del proyecto
|- README.md                # Descripción del TP e instrucciones de uso
```

Esta estructura es flexible. Se pueden agregar o eliminar archivos según sea necesario, pero es obligatorio incluir un Jupyter Notebook con todas las respuestas del TP.