UNIVERSIDAD SIMÓN BOLÍVAR

Departamento de Ciencia de la Información

CI-5438 - Inteligencia Artificial II

Profesora: Ivette Martinez

PROYECTO 1

Elaborado Por:

Edward Fernández 10-11121
Carlos Ferreira 11-10323
Stefani Castellanos 11-11394

Sartenejas, Febrero 2017

PLANTEAMIENTO DEL PROBLEMA

En la actualidad, Inteligencia Artificial es uno de los campos más estudiados en Ciencias de la Computación, por esta razón es de suma importancia comprender las técnicas básicas de uno de los subcampos de la misma; *Machine Learning*. Su objetivo es desarrollar técnicas que permitan a las computadoras aprender sobre un conjunto de datos.

Existen tres tipos de aprendizaje: supervisado, no supervisado y por reforzamiento; este proyecto se enfocará en el primero, utilizando Regresión Lineal Múltiple para ajustar una recta a un conjunto de datos dado y previamente etiquetado que permita hacer una predicción sobre una característica (y). Se utilizará una de las técnicas más ampliamente estudiadas y aplicadas en este campo: Descenso del Gradiente; que permite encontrar el mínimo de una función de costo ($J(\theta)$) actualizando los valores (θi) y utilizando la tasa de aprendizaje (α) en cada paso para aproximarse al mismo.

La función de costo escogida es la típica: el error cuadrático medio, ya que es una función cuadrática que posee un mínimo global por lo que no importa la inicialización de los θi (y se inicializan en 0). Adicionalmente se utilizarán técnicas de minería de datos y estadística para limpiar y normalizar los datos.

ACTIVIDADES

1. Implemente el algoritmo de Descenso del Gradiente para resolver una Regresión Lineal Múltiple en el lenguaje de programación de su preferencia entre C, C++, Java o Python.

El algoritmo "Descenso del Gradiente" se encuentra en el archivo *linearRegression.py* fue implementado en Python haciendo uso de la librería *numpy* porque provee una implementación eficiente de matrices y funciones para operar las mismas. Adicionalmente, se utilizó la librería *matplotlib* para realizar los gráficos correspondientes para cada pregunta y la librería *pandas* para manipular el conjunto de datos referente a los precios de casas.

2. Pruebe su implementación del algoritmo con los siguiente conjuntos de datos simples: Peso corporal en función del peso de cerebro y Tasa de Mortalidad .

En ambos conjuntos de datos las diferencias de escalas presentadas en los valores produjeron como resultado que el algoritmo no converge debido a errores de precisión de la máquina, por esta razón fue necesario realizar una normalización utilizando:

$$xi = \frac{xi - \mu}{s}$$

donde :

xi = valor de la instancia i del rasgo x
μ = media del rasgo x
s = desviación estándar del rasgo x

a. Peso corporal en función del peso de cerebro. El conjunto de datos se encuentra disponible en el archivo dataset_x01.txt.

El algoritmo converge para una tasa de aprendizaje de 0.1 al cabo de 172 iteraciones. Como se observa en la Figura 1 la función de costo decrece con gran rapidez hasta aproximadamente la iteración 60 donde se mantiene casi constante hasta converger, lo que sugiere que con una tasa de aprendizaje mayor "Descenso del Gradiente" también llegará al mínimo pero con mayor rapidez.

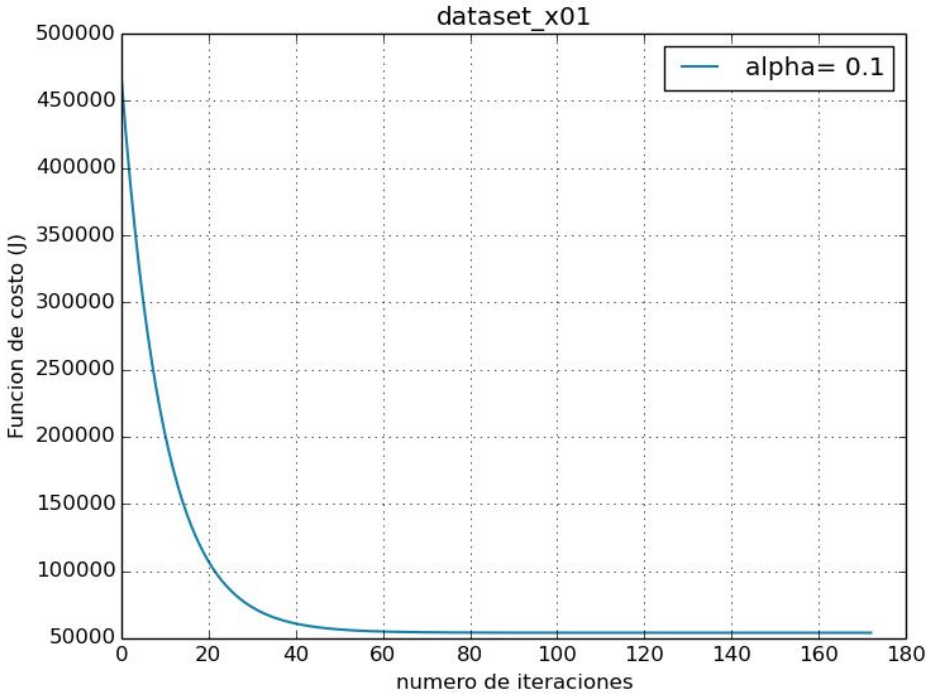

Figura 1. Función de costo vs número de iteraciones para el conjunto de datos peso corporal en función del peso de cerebro.

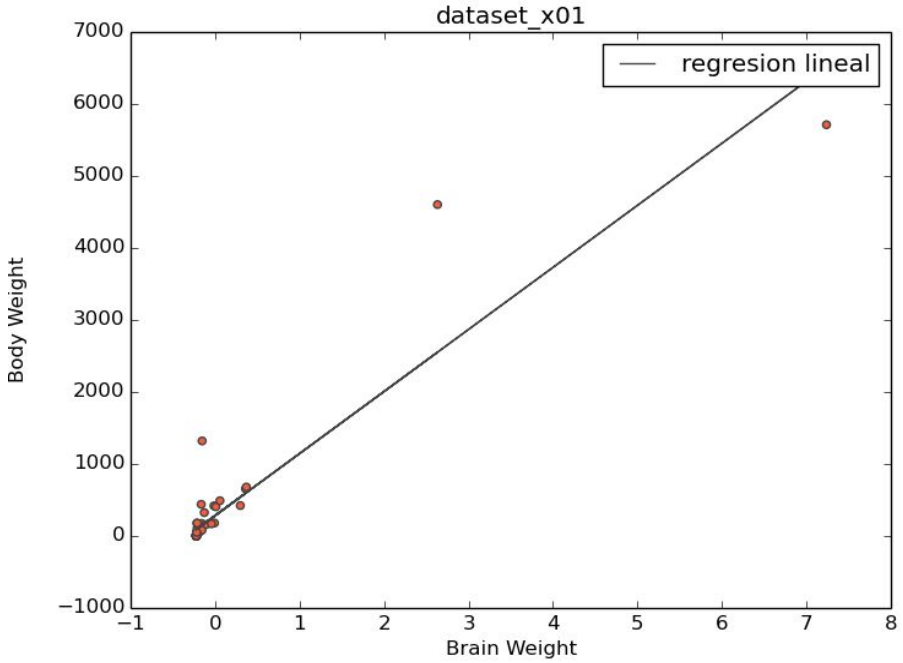

Figura 2. *Scatter plot* peso corporal en función del peso de cerebro. Cada punto representan los datos del conjunto y la línea gris es la función que mejor se ajusta según los resultados obtenidos por el Descenso del Gradiente.

b. Tasa de Mortalidad. El conjunto de datos se encuentra disponible en el archivo dataset_x08.txt.

Para este conjunto de datos se obtuve que converge incluso con una tasa de aprendizaje de 1.0 después de 20 iteraciones. En la figura 3 se pueden observar las curvas correspondientes a cada uno de los valores de *alpha* donde el valor 0.1 es el que presenta una convergencia más lenta (más de 100) iteraciones ya que cada paso que se da para descender por el gradiente es muy pequeño y le lleva más tiempo encontrar el mínimo.

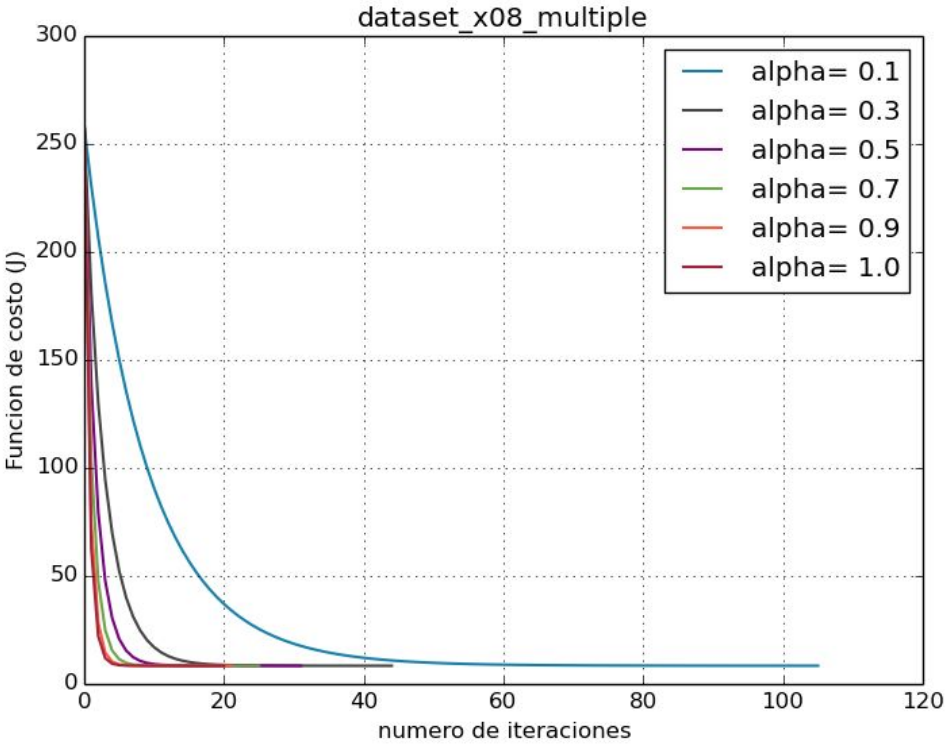

Figura 3. Función de costo vs número de iteraciones para el conjunto de datos de la tasa de mortalidad utilizando diferentes valores de la tasa de aprendizaje.

3. Construya un modelo que incluya todas los rasgos de datos del conjunto *Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project* para predecir el precio de un bien raíz. Para esto deberá:
    a. Limpiar los datos, usando las instrucciones de artículo de DeCock.
    b. Normalizar los datos resultantes de la limpieza.
    c. Escoger el 80 % de los datos para entrenamiento, y dejar un 20 % para la prueba
    d. Evaluar los resultados de su modelo en sobre los datos de entrenamiento y sobre los de prueba usando las cuatro métricas propuestas en la sección 4.

El primer paso para utilizar el conjunto de datos es eliminar aquellas instancias con un dato faltante ya que no hacen ningún aporte a la solución. El segundo paso fue filtrar según lo sugerido por DeCock en su artículo: eliminar todas las instancias en donde "Sale condition" no fuese "normal" y en donde "living area" es mayor a 1500 metros cuadrados; evidentemente luego de esto, el rasgo "Sale Condition" no presenta valores diferentes por lo que también es eliminado. Adicionalmente DeCock sugiere filtrar aún más los datos con una condición que convenga; fue elegida "Lot area" mayor a 10000 porque era la que eliminaba más instancias. Luego de esta limpieza inicial, se observa que "utilities" presenta el mismo valor (AllPub) para todas las instancias, por lo que es eliminada. Finalmente, se elimina la columna con el "PID" (número que identifica la parcela) ya que es único para cada casa y no se necesita en la predicción. Estos ajustes fueron realizados directamente en la hoja de cálculo y no con un algoritmo por razones de simplicidad.

Al observar los datos provistos, es importante destacar que existen 23 variables nominales y 23 ordinales por lo que es necesario convertir las primeras en valores numéricos para poder utilizar Descenso del Gradiente; se crea una nueva variable (*dummy)* para cada valor posible del rasgo en cuestión que puede tomar los valores 0 ó 1 indicando si está o no presente en esa instancia. Por ejemplo: para el rasgo *Street* se tienen como posibles valores "Grvl" y "Pave" por lo que se crean nuevas columnas "Street_Grvl" y "Street_Pave" para indicar si la instancia posee o no estas características. Dicho procedimiento se hace utilizando la función que provee *pandas*, *get_dummies()*.

Con respecto a las variables nominales, la única que presentaba problemas era "MS SubClass" ya que la función *get_dummies()* no la reconocía como tal, pues las etiquetas eran números, por esta razón se procesaron las etiquetas mapeando su valor a una letra del abecedario en orden ascendente (020 es A, 030 es B y así sucesivamente), de esta manera la función crea las variables necesarias para este rasgo.

La partición de se realizó utilizando las primera 616 instancias para entrenamiento y el resto (154 instancias) para prueba, separándolos en archivos para poder usar este último conjunto en la fase de predicción más adelante.

Mediante experimentos se logró observar que para una tasa de aprendizaje de 0.3 Descenso del Gradiente la diferencia entre el costo de una iteración y la siguiente no disminuye uniformemente pues los pasos para llegar al mínimo son muy grandes y lo sobrepasan. Por esta razón se utilizó el valor 0.1, sin embargo la convergencia es sumamente lenta por lo que en la Figura 4 se muestran solo 1000 iteraciones que muestran la tendencia que tiene a converger.

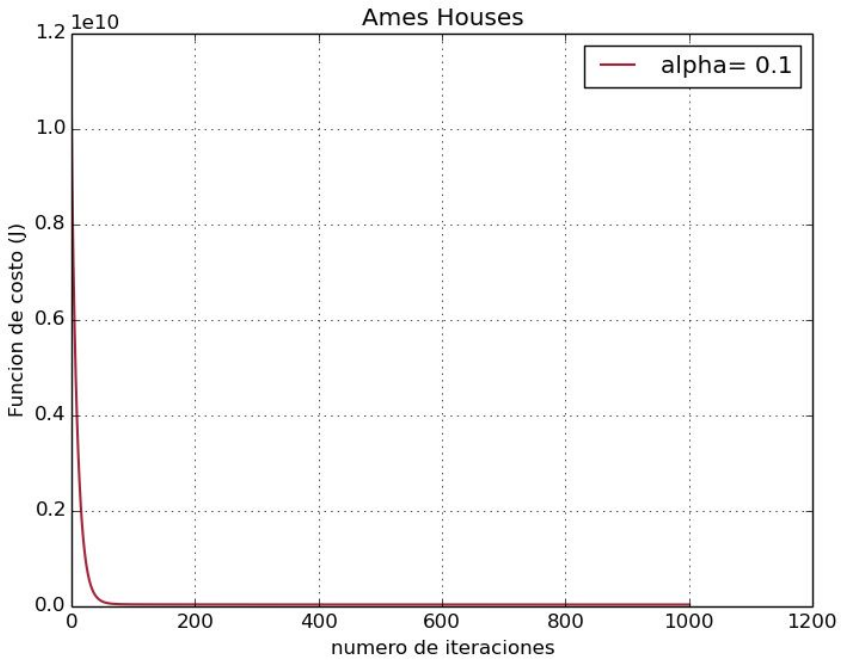

Figura 4. Función de costo vs número de iteraciones para el conjunto de datos Ames Housing

Para evaluar que tan buena es la predicción se utilizaron las métricas propuestas por DeCock:

- Sesgada (Bias): se calcula promediando el valor real menos el estimado. En la Figura 5 se aprecia que el modelo obtenido por el algoritmo subestima el valor real por 5000 $.

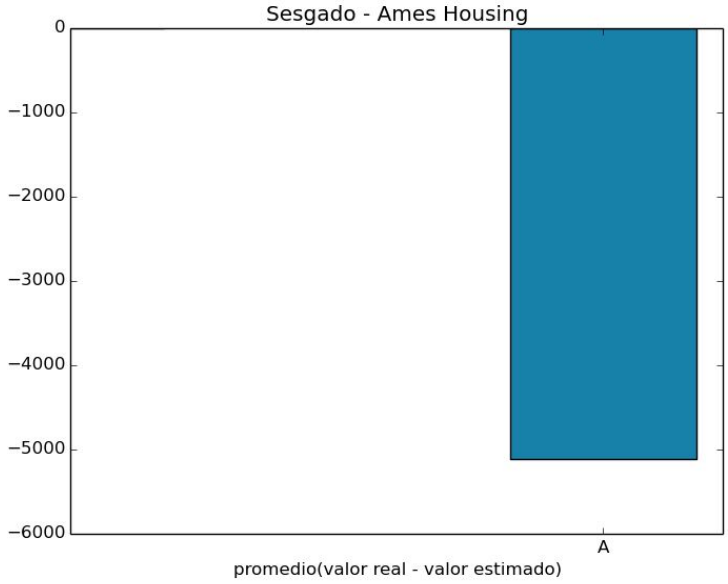

Figura 5. Métrica sesgada para el conjunto de datos Ames Housing

- Máxima desviación (maximum deviation): se calcula obteniendo la mayor diferencia entre el valor estimado y el real. Esta métrica permite concluir que el valor que peor estima el modelo se equivoca por 40.000 $

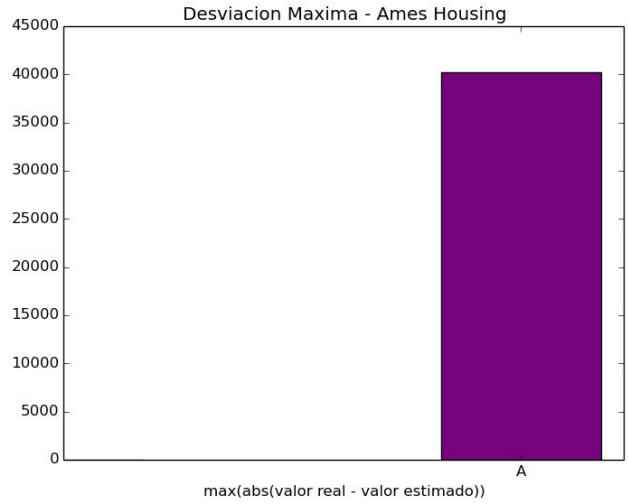

Figura 6. Métrica Máxima Desviación para el conjunto de datos Ames Housing

- Desviación absoluta media (Mean Absolute Deviation): se calcula con la media de la diferencia entre el valor estimado y el valor real.

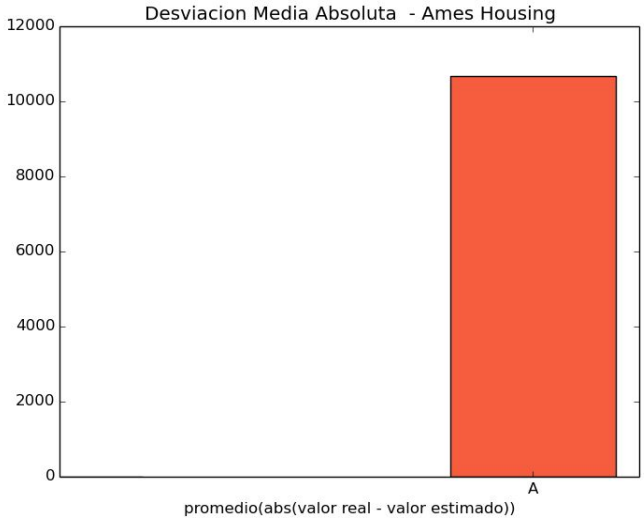

Figura 7. Métrica Desviación absoluta media para el conjunto de datos Ames Housing

- Error cuadrático medio (Mean Square Error): se calcula con la media de la diferencia entre el valor estimado y el valor real elevado al cuadrado.

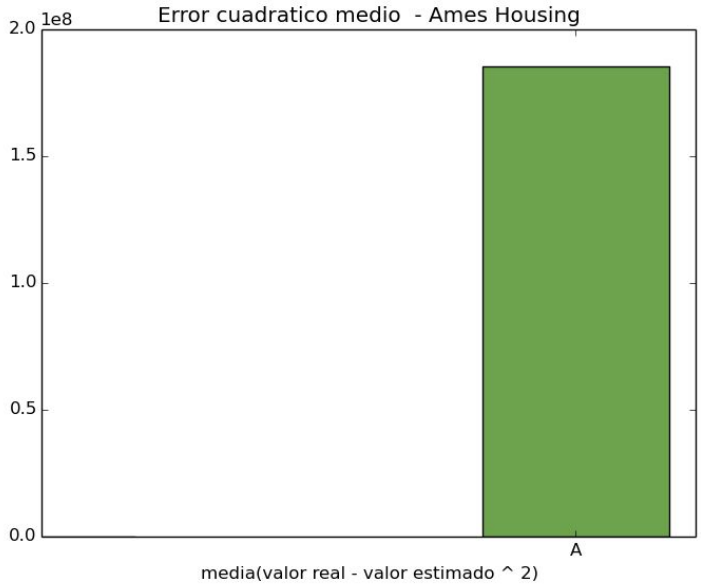

Figura 8. Métrica error cuadrático medio para el conjunto de datos Ames Housing

REFERENCIAS

De Cock, Dean Journal of Statistics Education, Volume 19, Number 3, (2011).

Russell, S. J., Norvig, P., & Canny, J. (2003). Artificial intelligence: A modern approach.