Práctica Regresión lineal y logística

En esta práctica de laboratorio, implementará la función de costo y los algoritmos de descenso de gradiente para la regresión lineal. La implementación debe estar en Python. Para ello, se proporciona un código de inicio. Consulte el archivo "LinearRegression.py" para obtener una base con la que comenzar. Los fragmentos de código que se espera que actualice están marcados con "AQUÍ USTED...".

Después de implementar las modificaciones requeridas, realice experimentos para responder las siguientes preguntas como parte de su reporte:

1. ¿Qué sucede si la tasa de aprendizaje es demasiado baja? Sugerencia: verifique diferentes valores para el grado de aprendizaje y analice su impacto

2. ¿Qué sucede si la tasa de aprendizaje es demasiado alta? Sugerencia: verifique diferentes valores para el grado de aprendizaje y analice su impacto

3. ¿Puede la regresión lineal realmente encontrar el mínimo global absoluto? Explique por qué/por qué no. Pista: puedes explicar el enfoque teórico y probarlo/refutarlo empíricamente

4. ¿Qué efecto tiene si cambia la suposición inicial de thet0 y thet1 para el descenso del gradiente a algo completamente fuera de lugar? Sugerencia: verifique diferentes valores para los valores theta y analice su impacto

5. ¿Qué sucede si no se actualiza thet0 y thet1 "simultáneamente" como debería, sino que se actualizan ambos parámetros en lazos separados (consulte el código)? Sugerencia: evalúelo empíricamente modificando el código en consecuencia

6. ¿Cuántas iteraciones del algoritmo de descenso de gradiente tiene que realizar para alcanzar los valores exactos correctos de theta0 y theta1? Pista: explique cómo lo calculó y demuéstrelo con resultados experimentales

7. Modifique su código para realizar la clasificación utilizando el conjunto de datos de diabetes que se incluye mediante la aplicación de **regresión logística**.

Realice y documente los experimentos. P.ej. modifique la tasa de aprendizaje, ejecute su código y vea qué sucede si este parámetro es muy pequeño. Además, proporcione un gráfico (pueden ser capturas de pantalla) para documentar el resultado de sus experimentos. No olvide proporcionar una breve explicación en forma de una o dos oraciones para que sepamos cuál es la causa de lo que muestra en la trama. También puede intentar generar datos lineales con algo de ruido y comprobar cómo afecta al proceso de entrenamiento y a los resultados.

Nota sobre el producto punto

Si tiene problemas al implementar el algoritmo de descenso de gradiente, revise cómo se implementa el producto punto en NumPy aquí:

https://numpy.org/doc/stable/reference/generated/numpy.dot.html

Además, las siguientes ecuaciones pueden ser útiles:

$$X = \begin{bmatrix} 1 & x_1^{(1)} \\ 1 & x_1^{(2)} \end{bmatrix}$$

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$$

$$\text{numpy.dot}(X, \theta) = \begin{bmatrix} \theta_0 * 1 + \theta_1 * x_1^{(1)} \\ \theta_0 * 1 + \theta_1 * x_1^{(2)} \end{bmatrix}$$