CSE643: Artificial Intelligence
Assignment-4

Divyajeet Singh (2021529)

November 25, 2023

Problem 1

(a)

We are given the equation for linear regression in five dimensions and the MSE loss in the expanded equation form and in vector form as follows

$$y = \mathbf{w}^\top \mathbf{x} + b = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + b \tag{1}$$

$$\mathcal{L}(\mathbf{w}; \{(\mathbf{x}^{(i)}, y^{(i)})\}) = \frac{1}{N}\sum_{i=1}^{N}(y_p^{(i)} - y^{(i)})^2 = \frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{w}^\top\mathbf{x}^{(i)} - y^{(i)}\right)^2 \quad i = 1, 2, \ldots, N \tag{2}$$

Deriving the gradients for linear regression using the expanded equation form, we get

$$\frac{\partial \mathcal{L}}{\partial w_j} = \frac{2}{N}\sum_{i=1}^{N}\left(\mathbf{w}^\top\mathbf{x}^{(i)} - y^{(i)}\right)x_j^{(i)} \quad j = 1, 2, \ldots, 5 \tag{3}$$

$$\frac{\partial \mathcal{L}}{\partial b} = \frac{2}{N}\sum_{i=1}^{N}\left(\mathbf{w}^\top\mathbf{x}^{(i)} - y^{(i)}\right) \tag{4}$$

Deriving the gradients for linear regression using the vector form, we get

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \frac{2}{N}\sum_{i=1}^{N}\left(\mathbf{w}^\top\mathbf{x}^{(i)} - y^{(i)}\right)\mathbf{x}^{(i)} \tag{5}$$

$$\frac{\partial \mathcal{L}}{\partial b} = \frac{2}{N}\sum_{i=1}^{N}\left(\mathbf{w}^\top\mathbf{x}^{(i)} - y^{(i)}\right) \tag{6}$$

Finally, we can also write the gradients in matrix form as follows (assuming $\mathbf{X}$ is the matrix of all inputs $\mathbf{x}^{(i)}$)

$$\nabla_{\mathbf{w}}\mathcal{L} = \frac{2}{N}\mathbf{X}^\top(\mathbf{X}\mathbf{w} - \mathbf{y}) \tag{7}$$

$$\nabla_b\mathcal{L} = \frac{2}{N}(\mathbf{X}\mathbf{w} - \mathbf{y})^\top\mathbf{1} \tag{8}$$

The equations above help us get the update rule for gradient descent. We first look at the update rule in the expanded equation form

$$w_j^{(t+1)} = w_j^{(t)} - \eta\frac{\partial \mathcal{L}}{\partial w_j} \quad j = 1, 2, \ldots, 5 \tag{9}$$

$$b^{(t+1)} = b^{(t)} - \eta\frac{\partial \mathcal{L}}{\partial b} \tag{10}$$

where η is the learning rate. We can also write the update rule in the vector form as follows

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{w}} \tag{11}$$

$$b^{(t+1)} = b^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial b} \tag{12}$$

Finally, we can also write the update rule in the matrix form as follows

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla_{\mathbf{w}} \mathcal{L} \tag{13}$$

$$b^{(t+1)} = b^{(t)} - \eta \nabla_{b} \mathcal{L} \tag{14}$$

In the above equations, the superscript t denotes the iteration number. The superscript $t+1$ denotes the next iteration.

(b)

In this problem, we were required to find the weight and bias for a regression problem using gradient descent implemented manually in Python. The code for the same is given in the file `DivyajeetSingh_gradient.py`. The code was run for 100 iterations as asked. The results are shown in the following figure.

```
PS C:\Users\Divyajeet Singh\Documents\College Documents\Semesters\Semester 5\Courses\Artificial Intelligence\CSE643> & "C:/Users/Divy
ajeet Singh/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/Divyajeet Singh/Documents/College Documents/Semesters/Semes
ter 5/Courses/Artificial Intelligence/CSE643/Assignment-4/DivyajeetSingh_gradient.py"
w = 22.98368393709005, b = 48.21294891107882
PS C:\Users\Divyajeet Singh\Documents\College Documents\Semesters\Semester 5\Courses\Artificial Intelligence\CSE643>
```

Problem 2

In this section, we were required to perform linear regression on the given dataset from the UCI-ML repository. The target is to achieve a threshold R^2 score.

(a) To handle the categorical features, we use one-hot encoding. The column `"Sex"` was one-hot encoded using the `pandas.get_dummies()` function.

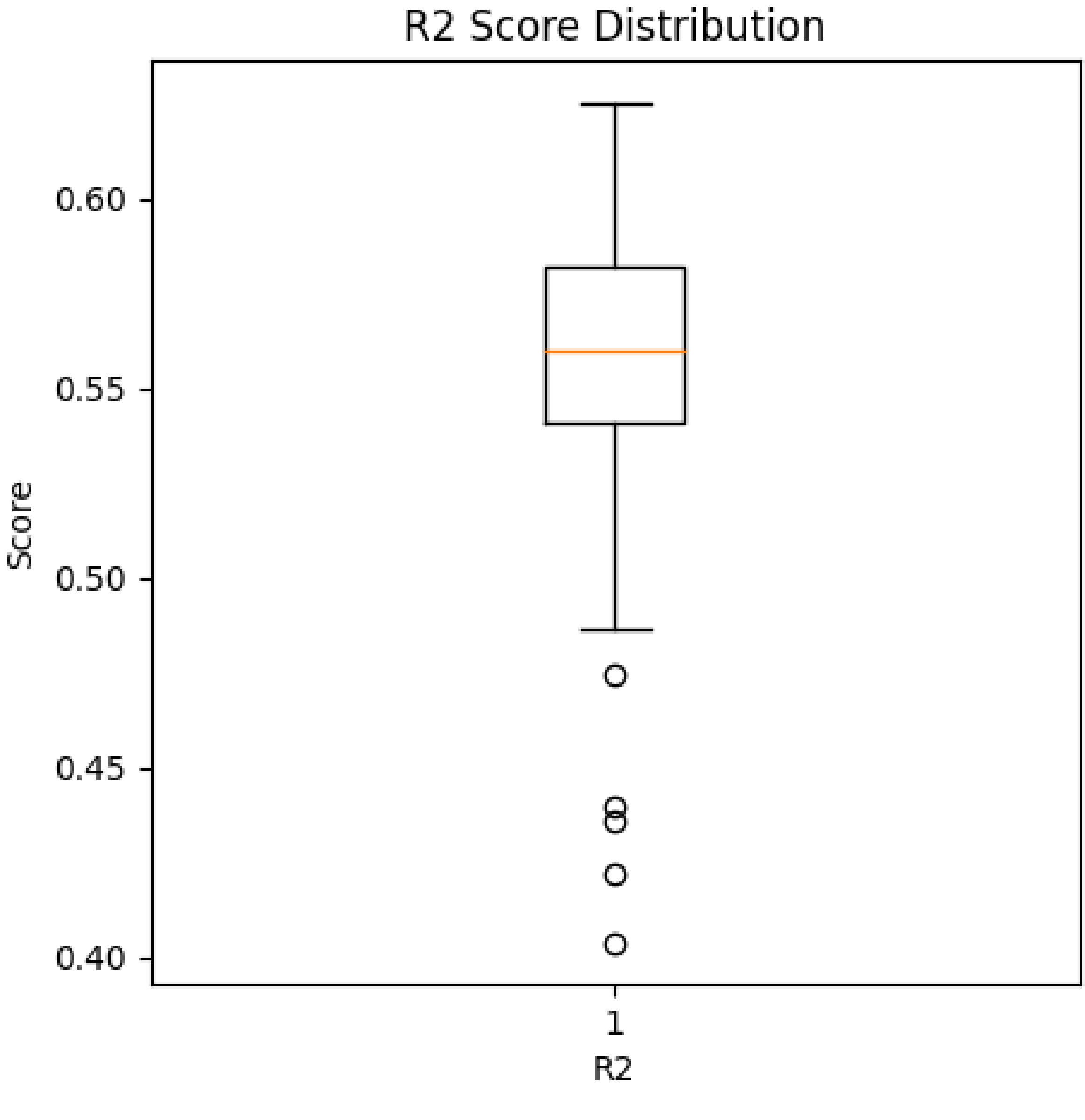
(b) As per the hint, a non-linear transformation was performed on the continuous features. The transformation used was the square function. The modified features were appended to the original dataset.

$$\texttt{new\_feature} \leftarrow \texttt{feature}^2 \tag{15}$$

The transformed features were used to train a linear regressor from the `sklearn` library. This achieved an R^2 score of 0.57 on the full dataset.

(c) 70-15-15 Cross Validation was performed for 100 iterations. The mean and standard deviation of the R^2 scores were calculated, which were 0.56 and 0.04 respectively.

The code for the same is given in the file `DivyajeetSingh_linear_regression.py`. The box plot for the R^2 scores is shown in the following figure.


R2 Score Distribution
0.60
0.55
0.50
0.45
0.40
Score
1
R2