Introduction to Machine Learning

Week 4 Assignment

Part 1:

1. Identify which type of machine learning (supervised, unsupervised, or re-inforcement learning) would be most appropriate for developing the smart thermostat.
   Answer: Re-inforcement learning

2. Explain why this type of learning is suitable for this application
   Re-inforcement learning is suitable because the smart thermostat algorithm will need to learn to make sequential decisions by interacting with the environment and receiving feedback.

Part 2

1.

Student Number	Actual Response	Predicted Value
1	**1**	**1**
2	**0**	**0**
3	**1**	**1**
4	**1**	**0**
5	**0**	**0**
6	**1**	**1**
7	**0**	**1**
8	**0**	**0**
9	**1**	**1**
10	**0**	**0**
11	**1**	**0**
12	**1**	**1**
13	**0**	**0**
14	**0**	**1**
15	**1**	**1**

- **True Positive (TP) occurs when Actual = 1 and Predicted = 1**
- **True Negative (TN) occurs when Actual = 0 and Predicted = 0**
- **False Positive (FP) occurs when Actual = 0 and Predicted = 1**
- **False Negative (FN) occurs when Actual = 1 and Predicted = 0**

	Positive (Predicted)	Negative(Predicted)
Positive (Actual)	**6 (TP)**	**2 (FN)**
Negative (Actual)	**2(FP)**	**5 (TN)**

2.

1. Accuracy = $\frac{\text{True Positives + True Negatives}}{\text{Total}}$

Accuracy = $\frac{6 + 5}{15} = \frac{11}{15}$ = **0.73 (73%)**

2. Precision = $\frac{\text{True Positives}}{\text{True Positives + False Positive}}$

Precision = $\frac{6}{6+2} = \frac{6}{8}$ = **0.75 (75%)**

3. Recall = $\frac{\text{True Positives}}{\text{True Positives + False Negative}}$

Recall = $\frac{6}{6+2} = \frac{6}{8}$ = **0.75 (75%)**

4. F1 Score = 2 . $\frac{\text{Precision . Recall}}{\text{Precision+ Recall}}$

F1 Score = 2. $\frac{0.75*0.75}{0.75+0.75}$ = 2. $\frac{0.5625}{1.5}$ = 2. 0.375 = **0.75 (75%)**

3. Yes. It is a fairly good model because:

1. In term of Accuracy the model correctly predicts about 73% of the instances.

2. In term of Precision the model correctly identified the positive class 75% of the time.

3. In term of Recall the model correctly identifies 75% of all actual "Yes" instances.

4. In term of F1 Score there is a balanced performance between precision and recall.

4. To improve the model, we can:

1. Increase Data Quality and Quantity

2. Cross-Validation

3. Hyperparameter Tuning

4. Handle Class Imbalance

5. Use Ensemble Methods

```
#IMPORT LIBRARIES
import numpy as np
import matplotlib.pyplot as plt
```

You are given a list of inputs x and outputs y and a linear regression model. The aim of this exercise is to validate the given model by calculating the mean square error (MSE) and the coefficient of determination (R-squared).

```
x=[-2.0,-1.6,-1.1,-0.67,-0.22,0.22,0.67,1.11,1.6,2.0]
y=[-0.46,1.3,0.27,1.4,2.6,2.9,3.1,2.3,3.8,3.6]
```

1. What is the design matrix X?

```
# CONSTRUCT THE DESIGN MATRIX
X= np.vstack((np.ones(len(x)), x)).T
```

2. The linear regression model is given by $y\_pre = \theta_0 + x\theta_1$, in matrix form we have $y\_pre = X * \theta$ where $\theta_0 = 2.08$ and $\theta_1 = 0.94$. Compute the predicted output y_pre

```
# CREATE A LIST FOR YOUR PARAMETERS
Theta = np.array([2.08, 0.94])

#COMPUTE THE PREDICTED OUTPUT
y_pre= X @ Theta
```

3. Compute the MSE and the R-squared

$$MSE = \frac{1}{n}\sum_{i=0}^{n}(y_i - y_{pred,i})^2$$

$$R^2 = 1 - \frac{SSR}{TSS}$$

$$SSR = \sum_{i=0}^{n}(y_i - y_{pred,i})^2$$

$$TSS = \sum_{i=0}^{n}(y_i - \overline{y_i})^2$$

```
MSE = np.mean((y - y_pre) ** 2)
print("Mean Squared Error:",MSE)
SSR = np.sum((y - y_pre) ** 2)
y_mean = np.mean(y)
TSS = np.sum((y - y_mean) ** 2)
r2 = 1 - (SSR / TSS)
print("R-squared:",r2)
```

```
Mean Squared Error: 0.34752241200000006
R-squared: 0.8079868920138196
```

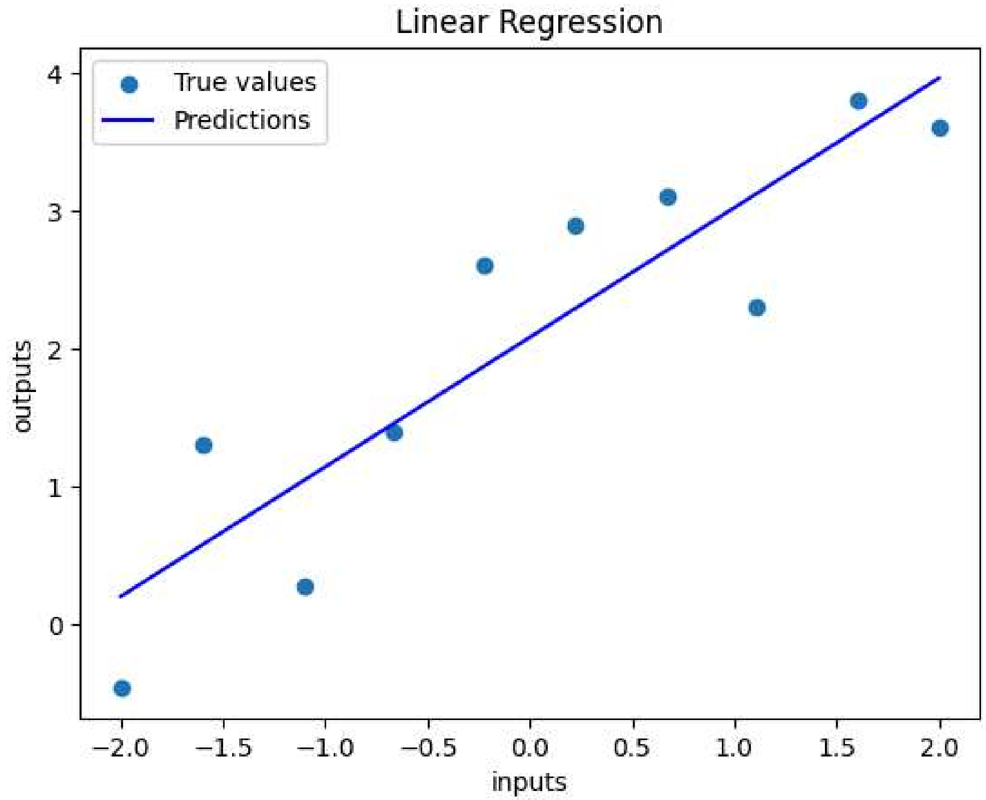
4. Is this model good?

Yes

5. Plot the true outputs and the predicted against the inputs on the same figure

```
#=======================
#Visualize the model
#=======================

plt.figure(1)
plt.scatter(x, y,label='True values')
plt.plot(x,y_pre, 'b',label='Predictions')
plt.xlabel('inputs')
plt.ylabel('outputs')
plt.title('Linear Regression')
plt.legend()
plt.show()
```



Start coding or generate with AI.